

Elias Haapamäki

VANHOJEN POTILASTIETOJEN ARKISTOINTI

Informaatioteknologian ja viestinnän tiedekunta
Diplomityö
Syyskuu 2019

TIIVISTELMÄ

Elias Haapamäki: Vanhojen potilastietojen arkistointi
Diplomityö
Tampereen yliopisto
Tietotekniikan diplomi-insinöörin tutkinto-ohjelma
Syyskuu 2019

Potilastietojärjestelmää vaihdettaessa joudutaan miettimään, mitä vanhoille potilastiedoille tehdään. Vaihtoehtoja ovat vanhojen tietojen siirtäminen uuteen järjestelmään tai vanhan järjestelmän jättäminen uuden rinnalle. Tietojen siirtäminen uuteen järjestelmään voi olla haastava ja pitkäkestoinen projekti ja siirron seurauksena saattaa syntyä virheitä. Vanhan järjestelmän jättäminen uuden rinnalle aiheuttaa ylimääräisiä kustannuksia ja pakottaa ylläpitämään vanhaa järjestelmää.

Kolmantena vaihtoehtona potilastietojärjestelmän tiedot voidaan siirtää Kelan kansalliseen potilastiedon arkistoon, minkä jälkeen Kela vastaa niiden säilytyksestä. Siirron jälkeen vanhan potilastietojärjestelmän ylläpidosta voidaan luopua ja tietoja voidaan katsella potilastiedon arkistosta. Sen jälkeen Kela vastaa myös aineiston tuhoamisesta säilytysajan loppumisen jälkeen. Tietojen siirto voi olla asiantuntijataitoja vaativa projekti, johon potilastietojärjestelmän ylläpitäjältä ei löydy osaamista.

Työn tavoitteena oli selvittää Atostekin potilastietojärjestelmien arkistointiprojekteihin liittyviä haasteita ja auttaa ymmärtämään projekteihin tarvittavia resursseja ja arvioimaan aikatauluja. Tavoitteena oli myös jäsentää ja dokumentoida arkistoinnissa käytetty prosessi, jotta sitä voitaisiin kehittää edelleen ja projektiin ryhdyttäessä voitaisiin paremmin arvioida projektin laajuutta. Prosessissa havaittujen ongelmien ratkaisu ei ollut työn varsinainen päämäärä, vaan niitä ratkaistaan myöhemmin.

Työssä tutkittiin Atostekin suorittamia potilastietojärjestelmien arkistointiprojekteja, joissa vanhan potilastietojärjestelmän tiedot siirrettiin kansalliseen potilastiedon arkistoon. Analysoitavaksi valittiin neljä eri projektia, jotka jaettiin kahteen ryhmään potilastietojärjestelmän aineiston toimitustavan mukaan. Ensimmäisessä ryhmässä aineisto toimitettiin poimimattomana eli siinä muodossa, jossa potilastietojärjestelmä aineistoa käsittelee. Toisen ryhmän projekteissa aineisto toimitettiin poimittuna, eli valmiiksi helposti analysoitavaan muotoon jäsennettyinä. Projekteja käsiteltiin datamigraatioprosesseina, jotka koostuivat aineiston toimituksesta, alkuanalyysistä, iteratiivisesta transformaatiovaiheesta, testauksesta ja validoinnista sekä tuotantoonviennistä. Työssä tutkittiin vaiheisiin liittyviä ongelmia ja haasteita ja selvitettiin kuinka ne vaikuttavat prosessin keston.

Työssä havaittiin, että potilastietojärjestelmien arkistointi voi olla haastava projekti, jonka kesto on vaikea arvioida etukäteen. Projekteihin voi liittyä yllätyksiä, joihin voidaan varautua tarkalla alkuanalyysillä. Poimittua aineistoa ei tarvitse analysoida, mutta silloin joudutaan luottamaan, että poiminta on tehty huolella, eikä aineistoa päästä vertaamaan lähdejärjestelmään. Aineiston poimiminen lähdejärjestelmästä voi olla hyvin haastavaa eikä aina onnistu. Myös virheiden mahdollisuus kasvaa, kun analysoija ei tunne järjestelmän toimintaa. Tuloksilla ei voida suoraan arvioida alkavan projektin kestoa, mutta ne auttavat varautumaa eri tilanteisiin ja huomioimaan projektin analyysivaiheessa tärkeitä asioita.

Avainsanat: datamigraatio, potilastietojärjestelmä, takaisinmallinnus.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Elias Haapamäki: Vanhojen potilastietojen arkistointi
Master of Science Thesis
Tampere University
Master's Degree Programme in Information Technology
September 2019

When changing the patient information System, one must think about what to do with the old medical reports. The options are to transfer the old data to the new system or to leave the old system side by side with the new one. Transferring data to a new system can be a challenging and time-consuming project and may result in errors. Leaving the old system alongside the new one will incur additional costs and force you to maintain the old system.

The third option is to transfer the information from the patient information system to Kela's (The Social Insurance Institution of Finland) national patient information archive, after which Kela is responsible for storing it. After the migration, the maintenance of the old patient information system can be stepped down and the information can be viewed from the Patient Data Repository. After that, Kela will also be responsible for the destruction of the material after the storage period has expired. Data migration can be a project that requires expert knowledge for which the patient information system administrator has no expertise.

The purpose of this work was to address the challenges of Atostek's patient information system archiving projects, and to help understand the resources needed for projects and evaluate schedules. The aim was also to structure and document the process used for archiving so that it could be further developed, and the scope of the project better evaluated when starting a project. Solving the problems encountered in the process was not the actual goal of the work but can be solved later.

The work analyzed Atostek's patient information system archiving projects, where old patient information system data was transferred to the national Patient Data Repository. Four different projects were selected for analysis, which were divided into two groups according to the delivery method of the patient information system data. In the first group, the data was delivered in a raw form, that is, in the form in which the data is processed by the patient information system. In the second group of projects, the material was delivered in a structured, ready-to-analyze format. The projects were treated as data migration processes consisting of data delivery, initial analysis, iterative transformation, testing and validation, and production. The work investigated the problems and challenges related to the phases and how they influence the duration of the process.

It was discovered that archiving patient information systems can be a challenging project, the duration of which is difficult to estimate in advance. Projects can come with surprises that can be anticipated with a thorough initial analysis. There is no need to analyze the extracted data, but one must trust that the extraction has been done carefully and the extracted data cannot be compared with the data in the source system. Extracting data from a source system can be very challenging and not always successful. Also, the possibility of errors increases when the analyzer is not familiar with the operation of the system. The work's results do not help in directly estimating the duration of the project, but they help to prepare for different situations and to consider important issues during the analysis phase of the project.

Keywords: data migration, patient information system, reverse engineering.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tahdon kiittää työnantajaani Atostek Oy:tä, josta sain sekä aiheen, että rahoituksen diplomityölle. Kiitän Atostekillä työtäni ohjannutta ja tarkastanutta Petteri Vigreniä kommentista ja kannustuksesta työn loppuun saattamiseen. Työn tarkastajana sekä ohjaajana toimi Timo Aaltonen, jota haluan kiittää työn tekoa ohjaavista kommenteista sekä palautteesta. Kiitän Atostekia kannustavan työilmapiirin luomisesta ja hyvästä työilmapiiristä. Kiitokset myös Ruutille kannustuksesta ja oikoluvusta.

Tampereella, 15.9.2019

Elias Haapamäki

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. DATAMIGRAATIO	3
2.1 Datamigraatio ja ETL-prosessi	3
2.1.1 Poiminta	5
2.1.2 Transformaatio	6
2.2 Potilastietojen arkistointi kirjallisuudessa	6
3. POTILASTIETOJEN ARKISTOINTI	10
3.1 Potilastietojen arkistointiprosessin ominaisuudet	11
3.1.1 Vaatimukset laista	11
3.1.2 Kanta-palveluiden asettamat vaatimukset	12
3.1.3 Tekniset vaatimukset	14
3.2 Vanhojen potilastietojen arkistointi -prosessi	17
3.2.1 Aineiston toimitus	20
3.2.2 Analyysiympäristön pystyttäminen	21
3.2.3 Arviointi ja analyysi	22
3.2.4 Transformaatio	24
3.2.5 Testaus	25
3.2.6 Validointi	26
3.2.7 Tuotantoonvienti	27
4. PROJEKTIN ESITTELY	28
4.1 Projektin A (Poimimaton aineisto)	28
4.2 Projektin B (Poimimaton aineisto)	29
4.3 Projektin C (Valmis aineistopoiminta)	30
4.4 Projektin D (Valmis aineistopoiminta)	31
5. VANHOJEN POTILASTIETOJEN ARKISTOINTI -PROSESSIN HAASTEET	32
5.1 Aineiston toimitus	32
5.2 Tietokannan analysointi	33
5.3 Järjestelmän analysointi	36
5.4 Tiedon luokittelu ja rikastus	38
5.5 Transformaatio	39
5.6 Tuotantoonvienti	40
5.7 Testaus	41
5.8 Muut ongelmat	42
5.9 Yhteenveto ongelmista	43
6. ONGELMAN RATKAISU	46
6.1 Alkuanalyysin kehittäminen	46
6.2 Testauksen kehittäminen	48
6.3 Datamigraation kattavuus	49
6.4 Sisäisen tietomallin käyttö	50

7.YHTEENVETO.....	53
LÄHDELUETTELO.....	57

1. JOHDANTO

Jokaisesta lääkärikäynnistä lääkäri kirjaa merkinnän potilasasiakirjaan. Merkintöjä syntyy myös vuodeosastolta, laboratoriokokeista ja puhelinsoitoista. Nämä tiedot on laissa nimetty *potilastiedoiksi*. Potilastiedot ovat lain [1] mukaan tietoja, jotka sisältyvät potilasasiakirjoihin. Potilasasiakirjat tarkoittavat ”*potilaan hoidon järjestämisessä ja toteuttamisessa käytettäviä, laadittuja tai saapuneita asiakirjoja taikka teknisiä tallenteita, jotka sisältävät hänen terveydentilaansa koskevia tai muita henkilökohtaisia tietoja*” [2].

Potilastietoa koskee määräys [3] säilytysajoista, jonka nojalla potilastietoa on säilytettävä. Tämä johtuu siitä, että potilastiedolla on merkittävä vaikutus potilaan tulevaan hoitoon. Säilyttäminen liittyy myös potilaan oikeusturvaan potilasvahinkojen tapauksessa sekä laadukkaan hoidon takaamiseen.

Tähän asti terveydenhuollon palveluntarjoajat ovat joutuneet ylläpitämään vanhoja järjestelmiään tai suorittamaan kalliita migraatioita, kun potilastietojärjestelmiä on vaihdettu. Ei ole lainkaan epätavanomaista, että uuteen järjestelmään ei ole mahdollista helposti siirtää tietoja vanhoista järjestelmistä ja lisäksi *datamigraatioon*, eli tietojen siirtämiseen vanhasta järjestelmästä uuteen järjestelmään, liittyy merkittävä epäonnistumisriski [4]. Tästä johtuen vanha järjestelmä voidaan joutua jättämään katselukäyttöön uuden järjestelmän rinnalle.

Kahden järjestelmän ylläpidosta saattaa seurata merkittäviä kustannuksia ja lisäksi se aiheuttaa käyttäjille ylimääräistä vaivaa, kun hoidossa vaaditaan toisen järjestelmän käyttöä. Esimerkiksi lääkemääräysten tarkistamiseksi, lääkärin on tarkastettava molemmista järjestelmistä potilaan nykyinen lääkitys. Kahden järjestelmän käyttö saattaa aiheuttaa ristiriitaisuusongelmia, jos toisesta järjestelmästä muokataan tietoa, mutta sitä ei päivitetä toiseen.

Ratkaisuna näihin ongelmiin vuoden 2015 jälkeen terveydenhuollon palveluntarjoajien on ollut mahdollista siirtää vanhat potilastiedot *kansalliseen potilastiedon arkistoon*, jonka jälkeen vanhasta järjestelmästä voidaan luopua. Tässä työssä tarkastellaan potilastiedon muuntamista potilastietojärjestelmästä kansalliseen potilastiedon arkistoon sopivaan muotoon. Arkistolaitoksen asetuksen mukaan Kela on nyt ainoa sähköisen potilastiedon pitkäaikaisarkistoinnin suorittaja. Tämä tarkoittaa, että terveydenhuollon palvelun antajat eivät voi säilyttää sähköisiä potilastietojaan missään muualla pitkäaikaisesti.

Siirtymäajan jälkeen terveydenhuollon palveluntarjoajat ovat velvoitettuja liittymään kansallisiin terveydenhuollon tietopalveluihin, jonka jälkeen arkistointia ei enää tarvita. Suomessa on kuitenkin todella paljon vanhoja järjestelmiä, jotka sisältävät arkistoitavia potilastietoja.

Potilastietojen arkistoinniseksi Kela tarjoaa palveluna *vanhojen potilastietojen arkistointia*. Sen käyttäminen on ainoa vaihtoehto, jos organisaatio haluaa luopua vanhojen tietojen säilytyksestä. Potilastietojärjestelmän tiedot ovat hyvin erilaisessa muodossa kuin potilastiedon arkistoon ladattavat tiedot. Tietojen muuntaminen sopivaan muotoon on yhdenlainen datamigraatioprosessi ja vaatii paljon teknistä osaamista, jota terveydenhuollon palveluntarjoajilta ei välttämättä löydy. Tällöin arkistointipalvelun käyttämisessä voidaan tarvita ulkoista apua.

Tämän työn tarkoituksena on tarkastella neljää toteutettua vanhojen potilastietojen arkistointiprojektia, jotka perustuvat Kelan tarjoamaan vanhojen potilastietojen arkistointipalveluun. Työssä dokumentoidaan projekteissa käytettävä prosessi, jolla projektit on tähän mennessä suoritettu ja pyritään kehittämään prosessia paremmin määritellyksi. Työssä verrataan käytettyä prosessia perinteiseen datamigraatioprosessiin. Lisäksi tavoitteena on arvioida prosessiin liittyviä riskejä, jotta projektin kestoa ja resurssitarpeita voidaan arvioida tarkemmin. Prosessin arvioinnin seurauksena, myös arkistoinnin laatu voi parantua, mikä voi auttaa, kun tietoja käytetään myöhemmin hoitotyössä.

Työssä ei käsitellä vanhojen potilastietojen arkistointiprosessissa käytettävää teknistä ohjelmistoratkaisua. Asiaa on aiemmin tarkasteltu toisessa diplomityössä [5]. Työssä käsitellään ainoastaan potilastietojen arkistointia kansalliseen potilastiedon arkistoon. Esimerkiksi sosiaalihuollon tiedoille on olemassa vastaava palvelu, johon sosiaalihuollon asiakirjat voidaan arkistoida. Sosiaalihuollon vastaava palvelu on tekniseltä toteutukseltaan hyvin erilainen verrattuna terveydenhuollon arkistointipalveluun, joten työssä käsiteltäviä prosessia ei voida suoraan käyttää sosiaalihuollon asiakirjojen arkistoinnissa.

Työ alkaa yleisen datamigraatioprosessin esittelystä *ETL*-prosessin avulla. Samassa luvussa esitellään potilastietojen datamigraatiosta kertovaa kirjallisuutta ja tutustutaan potilastietojen datamigraatioissa aiemmin esiintyneisiin ongelmiin. Sen jälkeen luvussa 3 tutustutaan potilastietojen arkistoinnin laista tuleviin ja teknisiin vaatimuksiin. Vaatimusten käsittelyn jälkeen luvussa kuvataan vielä potilastietojen arkistointi -prosessin ja sen vaiheet.

Luku 4 esittelee työn lähtökohtana olleet projektit. Projektien esittelyn jälkeen luvussa 5 käsitellään prosessin haasteita. Luvussa 6 esitellään ratkaisuja luvun 5 haasteisiin ja ongelmiin.

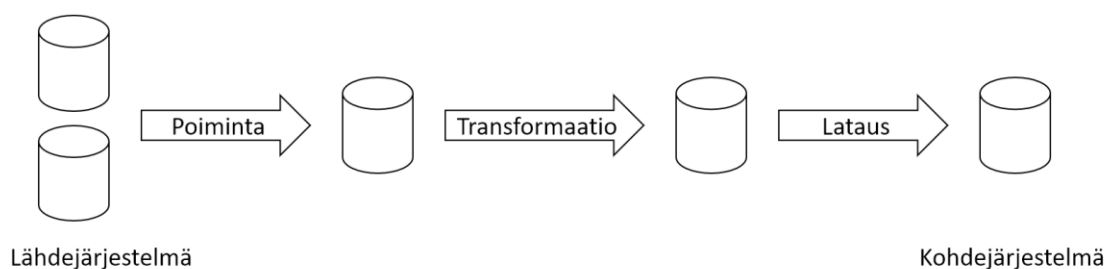
2. DATAMIGRAATIO

Potilastietojen datamigraatio vanhasta järjestelmästä uuteen tulee eteen, kun potilastietojärjestelmää vaihdetaan. Vaihto voi tapahtua saman järjestelmän versioiden välillä tai täysin uuteen järjestelmään siirryttäessä. Versiopäivityksessä datamigraatio on yleensä yksinkertainen, sillä datamigraatiossa voi olla mukana järjestelmän toimittaja ja toimittajalla on datamigraatiosta kokemusta. Uuteen järjestelmään siirryttäessä datamigraatiossa voi olla mukana vanhan ja/tai uuden järjestelmän toimittaja tai ulkopuolinen toimija. Seuraavaksi käsitellään datamigraatioprosessi ja mitä potilastietojen datamigraatioprosesseista on kirjoitettu kirjallisuudessa.

2.1 Datamigraatio ja ETL-prosessi

Teknologian kehittyminen ja sovellusalueiden tietämyksen lisääntyminen vaativat vanhojen tietojärjestelmien modernisointia, jotta niistä olisi hyötyä tulevaisuudessa [6]. Muutoksen tarve voi tulla esimerkiksi yrityskaupasta tai paremman yritysanalytiikan tarpeesta. Yritysten toiminnan kannalta kriittisten tietojärjestelmien modernisointi on kuitenkin haastavaa muun muassa seuraavien ongelmien takia: Järjestelmät toimivat vanhoilla laitteistoilla, ohjelmistojen ylläpito on yleensä kallista ja virheiden selvitys aikaa vievää, järjestelmien integrointi ei onnistu selkeiden rajapintojen puuttumisen takia ja vanhoja järjestelmiä voi olla mahdoton laajentaa [7].

Yksi vaihtoehto vanhan tietojärjestelmän modernisointiin on kehittää se alusta asti uudelleen nykyaikaisilla menetelmillä. Epäonnistumisen riski on kuitenkin yleensä liian suuri, joten tätä menetelmää käytetään harvoin. [7] Toinen vaihtoehto on kehittää järjestelmän ympärille *kääre* (engl. wrap), jolla vanhaa ohjelmistoa voidaan käyttää nykyaikaisin rajapinnoin ja järjestelmää voidaan laajentaa muuhun käyttöön. Kääre voidaan toteuttaa lisäämällä vanhaan järjestelmään laajennos, jota kääre voi käyttää. Jos vanhaa järjestelmää ei voida muokata, voidaan käärettä varten kehittää ohjelma, joka käyttää vanhaa järjestelmää interaktiivisesti [8]. Kolmantena vaihtoehtona vanhan järjestelmän tiedot ja toiminnallisuus voidaan siirtää uuteen nykyaikaisempaan ympäristöön. Siirtämistä kutsutaan migraatioksi ja seuraavaksi käsitellään tietojen siirtoa eli datamigraatiota.



Kuva 1. ETL-prosessi

Datamigraatiota voidaan kuvata ETL (Extract Transform Load) -prosessilla, jota on havainnollistettu kuvassa 1. ETL-prosessilla tarkoitetaan yleensä tietovarastojen koostamista yhdestä tai useammasta *lähdejärjestelmästä* tietovarastoon, siten että dataa muunnetaan ja siivotaan kohdejärjestelmään sopivaan muotoon. Yhden määritelmän [9] mukaan ETL on järjestelmä, joka irrottaa tietoja yhdestä tai useasta lähdejärjestelmästä yhteen kohteeseen muuntaen datan kohdejärjestelmään sopivaan muotoon ja varmistuen sen laadun.

ETL-järjestelmän tarkoituksena on koostaa tietoa hajanaisista lähtöjärjestelmistä yhteen tai useampaan kohdejärjestelmään. Tällä voidaan mahdollistaa vaiheittainen siirtyminen keskittyneisiin järjestelmiin esimerkiksi organisaation käyttäjänhallinnan osalta. ETL-prosessi on jatkuva prosessi, joka suoritetaan kerran ja sen jälkeen toistuvina eräajoina. Ensimmäisellä kerralla kaikki lähdejärjestelmän tiedot siirretään kohdejärjestelmään ja seuraavilla kerroilla siirretään vain muuttuneet tiedot.

ETL-järjestelmiä käytetään suurten tietovarastojen ylläpitoon. Nämä tietovarastot koostuvat useasta eri lähteestä kootusta tiedosta, joka yhdessä paikassa palvelee useaa käyttäjää paremmin. Koostamisen lisäksi ETL-järjestelmä parantaa datan laatua. Se poistaa virheitä ja korjaa virheellisiä arvoja, tarjoaa luotettavia arvioita datan luotettavuudesta, tallentaa transaktiot myöhempää käyttöä varten, sovittaa dataa useasta lähteestä käytettäväksi yhdessä sekä strukturoi dataa loppukäyttäjien työkaluja varten.

ETL-prosessi koostuu kolmesta vaiheesta. *Poimintavaiheessa* lähdejärjestelmistä irrotetaan halutut tiedot. *Transformaatiovaiheessa* irrotettu tieto muunnetaan kohdejärjestelmään sopivaan muotoon ja huolehditaan datan oikeellisuudesta ja yhteensopivuudesta muiden järjestelmien kanssa. Transformaatiossa voidaan esimerkiksi sovittaa vanhan potilastietojärjestelmän (lähdejärjestelmän) sukupuolen koodaus uutta potilastietojärjestelmää (kohdejärjestelmä) vastaavaksi. Lopulta irrotettu ja muunnettu tieto ladataan la-

taustavaiheessa kohdejärjestelmään. ETL ei koostu pelkästään edellä mainituista vaiheista vaan prosessiin kuuluu lisäksi myös testaaminen ja validointi, joita käsitellään potilastietojen arkistoinnin osalta aliluvuissa 3.2.5 ja 3.2.6.

2.1.1 Poiminta

Poimintavaiheen tärkein ongelma on vanhojen järjestelmien rajapintojen tunnistaminen. Lähdejärjestelmät voivat erota toisistaan tietokannan, käyttöjärjestelmän, laitteiston sekä tietoliikenneprotokollien osalta. Jokainen järjestelmä on yksilöllinen kokonaisuus, joka on huomioitava, jotta ETL-prosessi saataisiin vietyä läpi onnistuneesti [9].

ETL-prosessia käsittelevä kirja [9] suosittelee aloittamaan poimintavaiheen muodostamalla loogisen tietomallin, joka dokumentoi tietokenttien suhteet lähde- ja kohdejärjestelmien välillä. Tämän mallin avulla prosessin suunnitteluun saadaan lisää abstraktiota, mikä auttaa prosessin hahmottamisessa.

Loogisen tietomallin muodostaminen aloitetaan keräämällä lista tietolähteistä, jotka tietovarastoon viedään. Lähteiden sisältä valitaan oleelliset tiedot, joita analysoidaan tarkemmin tiedon profilointityökalulla. Työkalun tarkoituksena on varmistaa tiedon laatu, eli noudattaako tieto määrättyjä vaatimuksia, täydellisyys ja tarkoituksellisuus. Työkalulla saadaan tieto esimerkiksi ristiriitaisista tiedoista, kenttien virheellisistä arvoista tai puuttuvista arvoista. Kun tieto käydään läpi tässä vaiheessa, sen laatu ja hyödyllisyys saadaan varmistettua mahdollisimman varhaisessa vaiheessa.

Tutkimuksissa on havaittu, että tiedon profilointi- ja laadunvarmistus-työkalujen käyttöön otolla on ollut merkittävä vaikutus migraatioprojektien läpivientiin [10]. Jos työkalut kertovat, että tiedossa on puutteita, on muodostettava säännöt, joilla puutteet korjataan.

Poiminnan jälkeen tieto voidaan välivarastoida esimerkiksi tiedostoon. Kun tieto viedään irrotuksen jälkeen välivarastoon, myöhemmin tapahtuvan prosessoinnin aiheuttamasta virhetilanteesta voidaan selvittää aloittamalla prosessointi välivarastosta, eikä tietoa tarvitse irrottaa uudestaan. Hyöty korostuu erityisesti, jos irrotus on pitkäkestoinen tai lähdejärjestelmää kuormittava prosessi. Välivarasto mahdollistaa myös varmuuskopioiden ottamisen, sillä tiedoston voi pakata pieneen tilaan helpommin kuin tietokannan. Kolmantena hyötynä on auditoinnin mahdollistaminen. Kun tieto välivarastoidaan irrotuksen jälkeen, muunnosvaiheen auditointi eri vaiheiden välillä on helpompaa, sillä vertailu voidaan tehdä vaiheen sisäänmenon ja ulostulon välillä eikä ketjun alkupään ja vaiheen ulostulon välillä.

2.1.2 Transformaatio

Transformaatiovaiheessa lähtö- ja kohdetietomallit sovitetaan yhteen transformaatioiden ja kuvausten (engl. mapping) avulla. Tietoa suodatetaan ja puhdistetaan. Mallien yhteen sovittaminen ei muuta tiedon sisältöä, mutta suodattaminen ja puhdistaminen voi muuttaa. Puhdistamisessa tietoa standardoidaan ja ristiriitaisuudet käsitellään. Duplikaatit suodatetaan. Samaa tarkoittavat mutta eri käsitteellä kuvatut asiat yhdistetään. [9]

Transformaation tavoitteena on varmistaa, että luettu tieto on laadukasta. Laadukas tieto on

- virheetöntä
- yksiselitteistä,
- johdonmukaista,
- täydellistä.

Virheettömyys tarkoittaa, että tiedon sisällä ei ole ristiriitoja. Esimerkiksi henkilön osoitteen on sijaittava henkilön asuinkunnassa.

Yksiselitteisyys takaa, että tiedolla on vain yksi merkitys. Mikäli tiedolla on useita merkityksiä, siitä on siis selvítettävä mitä kyseinen tieto kyseisessä kontekstissa tarkoittaa. Esimerkiksi henkilöä ei voida yksilöidä pelkästään sukunimen avulla.

Johdonmukaisuus takaa, että samaa asiaa kuvataan aina yhdellä termillä. Esimerkiksi henkilön puhelinnumero on aina kentässä puh. eikä puhelinnumero.

Täydellisyydellä tarkoitetaan, että kaikilla tietueilla, joilla pitäisi olla arvo, on määritetty arvo. Tietoa ei saa puuttua eikä hävitä prosessin aikana.

Muunnosvaiheessa voi käyttää formaaleja tapoja [6] muunnoksen kuvaamiseen. Näitä kuvauksia arvioimalla voidaan tarkistaa, että prosessi suoritetaan oikein ja muunnoksen tarkistamiseen voi osallistua myös vähemmän teknisiä henkilöitä. Samassa tutkimuksessa osoitetaan myös tapoja transformaation validoinnille.

2.2 Potilastietojen arkistointi kirjallisuudessa

Taustakartoituksen tarkoituksena oli selvittää, minkälaisia haasteita potilastietojen arkistoinnissa tai datamigraatiossa on ilmennyt ja miten niihin voidaan varautua. Taustaselvitys aloitettiin etsimällä kirjallisuutta potilastietojen arkistoinnista tai migraatiosta, mutta kirjallisuutta näistä löytyi vain vähän. Aihetta laajennettiin potilastietojärjestelmän vaihtoon. Tutkimuksia siirtymisestä paperisesta potilastietojärjestelmästä sähköiseen [11]

löytyy useista aiheista kuten käytön helppoudesta lääkärin näkökulmasta ja potilasturvallisuusongelmista. Vastaavia tutkimuksia sähköisen potilastietojärjestelmän vaihtamisesta toiseen löytyy vähemmän ja vain muutamassa mainitaan mitä järjestelmien tiedoille tehtiin.

Yksi tapa, jolla tietoja voidaan siirtää, on tietojen syöttäminen käsin uuteen järjestelmään. Bentley ja muut seurasivat migraatiota [12], jossa potilastietojärjestelmän tiedot siirrettiin uuteen järjestelmään yhden viikonlopun aikana. Turvallisuushuolien takia tietojen siirto katsottiin lääkäreiden vastuulle, jotka kirjasivat osan 1000:n potilaan tiedoista käsin uuteen järjestelmään. Osa tiedoista, kuten laboratorio- ja kuvantamislausekkeet siirrettiin automaattisesti. Uudessa järjestelmässä potilaan tietoihin lisättiin merkintä tietojen onnistuneesta siirtämisestä, kun potilaan tiedot olivat kokonaan siirretty, jotta tiedettiin siirtänyt käyttämään vain uutta järjestelmää. Suuri osa siirrosta tapahtui 12 tunnin kuluessa ja loput saatiin siirrettyä vuorokauden aikana. Raportissa ei kerrota, aiheutuiko tietojen käsin syötöstä ongelmia tai vaaratilanteita. Tutkimuksista havaitaan, että käsin syöttämisen ongelmana on operaation pitkäkestoisuus ja/tai suuri ihmisresurssien tarve.

Käsin syöttämisessä on tärkeä varmistaa, että kaikki tarpeelliset tiedot siirretään. Girard [13] kertoo tapauksesta, jossa järjestelmän tiedoista siirrettiin vain osa ja tarkoituksena oli, että lääkärit päivittävät tiedot uuteen järjestelmään, kun niitä katsotaan vanhasta järjestelmästä. Näin ei kuitenkaan tapahtunut, vaan virheen seurauksena potilaalle määrättiin lääkettä, jolle hän oli allerginen ja tilanne johti vakavaan allergiseen reaktioon.

Käsin syöttäminen voi olla myös hidasta ja kallista. Vastaavassa migraatiossa [14] kerrottiin, että 45 000:n potilaan tietojen siirtäminen maksaisi palkkakustannuksissa noin \$60,000 - \$135,000. Huomattavasti suurempi ongelma olisi, että tietojen syöttö kestäisi vähintään puoli vuotta, työntekijöiden ajan puutteesta johtuen. Lisäksi puolen vuoden aikana tietoja kertyy lisää, jolloin siirtoja jouduttaisiin tekemään uudestaan.

Käsin syöttämisen lisäksi migraatio voidaan suorittaa automaattisesti [14]. Ohjelmisto voi lukea tietoja lähdejärjestelmän tietokannasta ja kirjoittaa niitä migraation kohdejärjestelmään tai syöttää tiedot kohdejärjestelmän käyttöliittymällä. Käsin syöttämisen sijaan, edellisessä tutkimuksessa tiedot päätettiin siirtää automaattisesti ohjelmoidulla tietojen syötöllä, jossa tiedot luettiin lähdejärjestelmän tietokannoista ja syötettiin myöhemmin ohjelmallisesti kohdejärjestelmään. Prosessi suoritettiin kolme kertaa, jotta prosessin aikana syötetyt uudet tiedot siirtyvät myös uuteen järjestelmään. Automaattisella tietojen syötöllä migraation tietojen vienti kesti vain 1,2 prosenttia käsin syöttämiseen kuluvaan aikaan verrattuna.

Gettinger ja muut seurasivat migraatiota [15], jossa siirryttiin sähköisestä potilastietojärjestelmästä uuteen potilastietojärjestelmään automaattisella konversiolla. Projektin järjestelmätoimittaja ei suositellut automaattista muunnosta muiden asiakkaiden kokemusten perusteella, vaan tietojen syöttämistä käsin. Projektitiimi kuitenkin katsoi, että käsin syöttämiseen menisi liikaa aikaa ja järjestelmissä on samankaltaisuuksia, joten muunnos päätettiin suorittaa automaattisesti.

Projektin alussa oletettiin, että migraatio olisi helppo ja siitä olisi suuri hyöty työntekijöille. Automaattinen muunnos ei kuitenkaan onnistunut täydellisesti kaiken tiedon osalta, vaan osa tiedoista jouduttiin syöttämään järjestelmään käsin. Ongelmia aiheuttivat esimerkiksi vanhan ja uuden järjestelmän erot lääkitysten ja tautiluokitusten koodauksen välillä. Ongelmien vuoksi vanhaa järjestelmää käytettiin vielä vuoden jälkeen projektin päättymisestä.

Täydellisen migraation lisäksi voidaan tyytyä myös ratkaisuun, jossa vanhat järjestelmät jätetään uusien järjestelmien rinnalle. Vanhat järjestelmät asetetaan vain-luku -tilaan, mutta niitä voidaan edelleen käyttää potilaan hoidossa. Bornstein [16] raportoi migraatiosta, jossa siirryttiin useasta järjestelmästä yhteen järjestelmään osittaisella automaattisella muunnoksella. Kaikkea tietoa ei ladattu automaattisesti, sillä pelättiin muotoiluerojen aiheuttamia ongelmia.

Makar [17] kertoo kolmesta migraatiosta, jossa vaihdettiin sairaalan sähköiset potilastietojärjestelmät uusiin. Tutkimuksessa havaittiin kolme tapaa suorittaa migraatio. Osa vanhoista järjestelmistä voidaan jättää rinnakkaiskäyttöön siirtymäajaksi, jolloin vanhoja tietoja voidaan katsoa vanhasta järjestelmästä ja tietoja voidaan syöttää uuteen järjestelmään. Toisessa menetelmässä vanhan järjestelmän tietoja voidaan siirtää uuteen tietokantaan, johon uusi järjestelmä ottaa yhteyden. Näin käyttäjä voi katsoa potilaan tietoja aivan kuin ne olisivat uudessa järjestelmässä. Myös tässä tutkimuksessa yhtenä vaihtoehtona oli tietojen manuaalinen syöttö. Menetelmää käytettiin yhdessä kahden edellisen vaihtoehdon kanssa tai täysin irrallisesti. Vaihtoehtona ei kuitenkaan mainittu automaattista tietojen siirtoa. Haastatteluiden perusteella automaattinen tietojen siirto koettiin liian haastavaksi ja haastateltu kertoi, että automaattinen tietojen siirto on haastavaa myös saman toimittajan tuotteiden välillä arkkitehtuurien erojen vuoksi.

Tutkimusten perusteella voidaan havaita, että automaattinen tietojen muuntaminen on haastava ongelma ja virheiden seurauksena voi olla potilasturvallisuutta vaarantavia tilanteita. Toisaalta prosessin kesto on huomattavasti lyhyempi manuaaliseen tietojen syöttämiseen verrattuna. Kaikille tutkimuksille oli yhteistä, että vain osa tiedoista valittiin siirrettäväksi ja tietoja saatettiin poimia myös esimerkiksi vain kahden edeltävän vuoden

ajalta. Vanhoja järjestelmiä jätettiin uusien järjestelmien rinnalle, sillä täydellinen tietojen ja ominaisuuksien siirtäminen ei aina onnistu.

3. POTILASTIETOJEN ARKISTOINTI

Vanhojen potilastietojen arkistointi on eräänlainen ETL-prosessi. Prosessin tavoitteena on koostaa tiedot useasta terveydenhuollon järjestelmästä yhteen keskitettyyn tietovarastoon, joten voidaan ajatella, että jokainen yksittäinen arkistointi on osa ETL-prosessia, jossa tiedot keskitetään yhteen tietovarastoon. Työssä myös yksittäinen arkistointi muodostaa oman ETL-prosessin, jossa tiedot poimitaan lähdejärjestelmästä ja ladataan kohdejärjestelmään.

Potilastietojen arkistointi -prosessi eroaa edellä käsitellystä migraatioprosessista käsiteltävien tietojen johdosta. Prosessille asettavat vaatimuksia Suomen laki ja EU:n asetukset sekä viranomaiset, joiden kanssa prosessi suoritetaan. Luvun tarkoituksena on kertoa vanhojen potilastietojen arkistoinnin prosessin erityispiirteet eli esitellä vaatimukset, jotka aiheutuvat laista ja määräyksistä.

Potilastietojen arkistoinnin erityispiirteenä on se, että arkistoinnin muoto on määritelty kansallisella eikä arkistojia voi itse vaikuttaa täysin tiedon lopulliseen muotoon. Kansalliset määrittelyt kattavat asiakirjojen metatiedot sekä sisällön esitysmuodon, mutta eivät ota kantaa asiakirjan sisältöön tai sen esitystapaan. Se tarkoittaa, että vaikka sisällön esitysmuoto on määrätty kansallisesti, jokainen toimija valitsee itse, kuinka sisältö muodostetaan. Potilastiedolle ei ole siis määritetty erillisiä rakenteita esimerkiksi mittausten tallentamiseksi.

Potilastietoja koskee luvussa 3.1.1 määritetty säilytysvelvollisuus. Säilytysvelvollisuudesta johtuen, potilastiedon haltijalla saattaa olla käytössä potilastietojärjestelmiä, joiden ainoa tarkoitus on mahdollistaa vanhojen tietojen säilytys ja käyttö. Työn projektien yhteydessä on havaittu, että Suomessa pienet toimijat jättävät vanhan potilastietojärjestelmän yleensä uuden järjestelmän rinnalle uuteen järjestelmään siirryttäessä, sillä tietojen siirto uuteen järjestelmään on haastavaa.

Potilastietojen arkistointi -prosessin tarkoituksena on mahdollistaa näistä vanhoista järjestelmistä luopuminen, jolloin vapaudutaan mahdollisista ylläpitokustannuksista sekä lisenssimaksuista. Järjestelmistä luopuminen on mahdollista ainoastaan, kun tietojen arkistointivelvoite saadaan täytettyä toisella tavalla. Kela mahdollistaa arkistointivelvoitteen täytön tarjoamalla vanhojen tietojen säilytystä.

Monimuotoisista potilastietojärjestelmistä johtuen, potilastietojärjestelmästä voi löytyä myös muuta kuin potilastietoa. Näitä tietoja koskevat potilastiedosta poikkeavat säilytys-

vaatimukset, joten järjestelmästä luopuminen on kokonaisvaltainen prosessi, jossa vanhaa järjestelmää on tarkasteltava kokonaisuudessaan. Näiden kaltaisia tietoja voivat olla esimerkiksi hallinnollinen tieto, kuten palkkaus tai kirjanpito. Tässä työssä keskitytään kuitenkin vain itse potilastiedon arkistointiin.

Arkistoinnin ainoa syy ei aina ole vanhoista järjestelmistä luopuminen, vaan arkistointi voidaan tehdä myös siksi, että tieto saadaan helpommin käyttöön organisaation sisällä. Vanhojen tietojen siirtäminen Kantaan mahdollistaa sen, että saman organisaation sisällä eri käyttäjät pääsevät näkemään helpommin potilaiden tietoja.

3.1 Potilastietojen arkistointiprosessin ominaisuudet

Seuraavissa aliluvuissa käsitellään laista ja muista lähteistä aiheutuvat potilastietojen arkistointiprosessin vaatimukset. Potilastietojen säilytysvelvoite ja velvollisuus huolelliseen käsittelyyn tulevat laista. Arkistoinnissa on mukana myös viranomaisia, jotka asettavat omat vaatimuksensa.

3.1.1 Vaatimukset laista

Laki asettaa potilastietojen käsittelylle vaatimuksia koskien esimerkiksi niiden säilytystä, hävittämistä ja salassapitoa. Potilastietojen säilytysajasta on määrätty potilasasiakirja-asetuksen liitteessä potilasasiakirjatyypeittäin [3]. Sosiaali- ja terveysministeriön opas potilasasiakirjojen laatimisesta ja käsittelystä kertoo säilytysajasta seuraavasti: *”Kun potilasasiakirja-asetuksen liitteessä tarkoitettu säilytysaika on päättynyt tai kun sen jälkeen säilytetyt potilasasiakirjat, näytteet ja elinmallit eivät enää ole välttämättömiä potilaan hoidon järjestämisen tai toteuttamisen kannalta, ne on hävitettävä välittömästi ja siten, että sivulliset eivät saa niistä tietoa.”* [18] Potilasasiakirjoja, näytteitä ja elinmalleja voi säilyttää myös vaaditun säilytysajan päättymisen jälkeen, mikäli katsotaan, että se on tarpeellista.

Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä [1] edellyttää julkisen terveydenhuollon palvelujen tarjoajan liittymistä valtakunnallisten tietojärjestelmäpalveluiden käyttäjäksi. Yksityisten palveluntarjoajien tulee myös liittyä, jos potilasasiakirjojen pitkäaikaissäilytys toteutetaan sähköisesti. Valtakunnallisen tietojärjestelmäpalvelun tehtävänä on muun muassa vastata terveydenhuollon ja sosiaalihuollon palvelujen tarjoajien tietojen säilytyksestä niiden puolesta. Ennen asetuksen voimaantuloa valmistuneet asiakirjat voidaan tallettaa valtakunnalliseen arkistointipalveluun. Liittymisen jälkeen valmistuneiden asiakirjojen alkuperäiskappaleet on tallennettava valtakunnalliseen arkistointipalveluun, eikä niitä tarvitse tallentaa muualle.

Arkistolaitos on päätöksellään [19] määrännyt, että kaikki arkistointipalveluun siirretyt vanhat tiedot (ennen arkistointipalveluun liittymistä syntyneet asiakirjat) säilytetään tois-
laiseksi pysyvästi yksiomaan sähköisessä muodossa eli niitä ei tulosteta paperille tai
mikrofilmille. Tiedot voidaan hävittää alkuperäisistä järjestelmistä siirtymisen onnistumi-
sen varmistamisen jälkeen. Sähköisissä järjestelmissä säilytettäviä tietoja ei tule hävittää
ennen kuin on varmistettu, voidaanko ne siirtää arkistopalveluun.

Potilastiedot kuuluvat EU:n tietosuoja-asetuksessa [20] niin sanottuun *erityiseen henki-
lötietoryhmään* eli *arkaluonteisiin tietoihin*. Arkaluontoisten tietojen käsittely on lähtökoh-
taisesti kielletty, mutta Suomen lainsäädännössä on kuitenkin säädetty poikkeuksia
muun muassa terveydenhuollon palvelujen toteuttamiseksi. Potilastiedot säilytetään po-
tilasrekisterissä, jonka käsittelystä on sovittava erikseen käsittelijän kanssa. Kun tietoja
käsitellään rekisterinpitäjän lukuun, käsittelijän ja rekisterinpitäjän on solmittava sopi-
mus, jolla varmistetaan, että rekisteröityjen oikeudet toteutuvat myös muiden käsitel-
lessä tietoja.

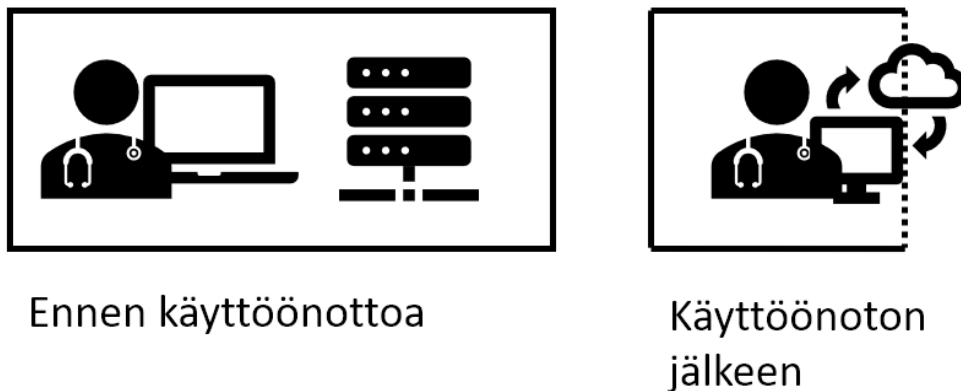
Laki vaikuttaa potilastietojen arkistointiin mahdollistaen niiden arkistoinnin ja asettamalla
vaatimuksia tietojen käsittelylle. Tietojen käsittelyssä tulee huolehtia tietosuoja-asetuk-
sen täyttymisestä ja salassapidettävien tietojen turvaamisesta.

3.1.2 Kanta-palveluiden asettamat vaatimukset

Edellisessä luvussa mainittiin *Potilastiedon arkisto* (Arkisto) ja valtakunnallinen tietojär-
jestelmäpalvelun arkistointipalvelu, joista edellinen toteuttaa jälkimmäisen. Arkisto on
osa *Kanta-palveluita*, jotka ovat lainsäädännön [1] vaatimusten toteuttamiseksi raken-
nettu palvelukokonaisuus, josta vastaavat useat kansalliset toimijat. Potilastiedon arkisto
on kansallinen terveydenhuollon tietojärjestelmä, joka mahdollistaa keskitetyn sähköisen
potilastietojen arkistoinnin ja käytön sekä tietojen pitkäaikaisen säilyttämisen. Arkistolle
ei ole omaa käyttöliittymää vaan se tarjoaa rajapinnat, joilla potilastietojärjestelmät voivat
käyttää sen palveluita. Siellä säilytetään sekä uudet että vanhat potilastiedot. Kansalai-
set pääsevät katsomaan uusia tietoja *Omakannan* [21] avulla, mutta vanhat tiedot ovat
vain terveydenhuollon ammattilaisten nähtävillä. Potilaalla on oikeus määrätä tietojensa
käytöstä siten, että ilman erillistä lupaa, tietoja voivat käyttää vain rekisterinpitäjän orga-
nisaatio. Myös tämä *suostumusten hallinta* on Kelan tuottama palvelu.

Vanhat potilastiedot tarkoittavat tietoja, jotka ovat tallennettu terveydenhuollon tietojär-
jestelmiin tai paperiarkistoihin ennen kuin organisaatio on liittynyt Potilastiedon arkistoon
[22]. Lisäksi vanhoja potilastietoja ovat tiedot, joita Potilastiedon arkiston käyttöönoton
yhteydessä ei ole vielä voitu arkistoida STM:n (Sosiaali- ja terveysministeriö) vaihease-
tuksen vuoksi.

Verrattuna datamigraatioon, jossa tiedot siirretään asiakkaan vanhasta järjestelmästä uuteen järjestelmään, vanhojen potilastietojen arkistoinnissa siirrettävien tietojen kohteena on kansallinen potilastiedon arkisto, jota asiakas ei hallitse. Potilastiedon omistaja ei voi arkistoida tietoja arkistoon suoraan, vaan arkistoijan on käytettävä Kanta-palveluihin kuuluvaa Kelan tarjoamaa potilastietojen arkistointi -palvelula, jolla vanhat potilastiedot viedään Potilastiedon arkistoon pysyvässä säilytyksessä.



Kuva 2. Käyttö Arkiston käyttöönoton jälkeen.

Kuva 2 esittää järjestelmän käytön erot ennen ja jälkeen Arkiston käyttöönoton. Vasemmalla on tilanne, jossa tiedot sijaitsevat asiakkaan omissa järjestelmissä ja oikealla tilanne, jossa tiedot haetaan Arkistosta. Oikean kuvan pilvi kuvaa keskitettyä tietojärjestelmää eli Arkistoa, jossa potilastietoja säilytetään.

Vanhojen potilastietojen arkistointi on Kelan tarjoama palvelu, jossa potilastietojärjestelmien vanhat tiedot voidaan arkistoida pitkäaikaisesti Potilastiedon arkistossa. Palvelu sisältää potilastietojen massalatauksen potilastiedon arkistoon, arkistoinnin palvelurajapintojen kautta, asiakirjojen kuvailutietojen haun ja asiakirjojen noudon palvelurajapinnan kautta. [22]

Potilastietojen arkistointiin vaikuttaa myös THL (Terveystieteiden tutkimuskeskus), joka määrittelee Potilastiedon arkistossa käytettävät tietosisällöt ja määrittelyt. THL määrittää Kanta-palveluiden toteuttamisen edellyttämät ja potilastietojen arkistoinnissa vaadittavat tietosisällöt, käsittemallit ja toimintaprosesseja tukevat tietorakenteet. Tietosisällöt julkaistaan kansallisessa koodistopalvelussa koodistoina. Koodisto on määritelty THL:n koodistopalvelun sanastossa [23] seuraavasti: *"Koodisto on tiettyyn käyttöön valmistettu tietokokonaisuus, joka muodostuu yksittäisistä määritellyistä koodeista ja koodistoon liittyvistä metatiedoista (kuvailutiedoista) - -"*. Koodi on: *"- - tiettyä käsitettä vastaava objekti*

(kohde tai olio), jonka ominaisuuksia (attribuutteja) ovat koodistopalvelinmuodossa mm. koodin tunnus, koodin nimi, koodin kuvaus ja koodistoviittaus.”

Potilastietojen arkistoinnissa on siis huomioitava THL:n määräykset koskien Potilastiedon arkistoa, sekä Kelan määräykset potilastietojen arkistoinnin osalta. THL pitää myös seurantakokouksia, joilla seurataan projektien kulkua ja ratkaistaan ongelmia.

3.1.3 Tekniset vaatimukset

Edellä käytiin läpi yleinen kuvaus vanhojen potilastietojen arkistointi -palvelun sisällöstä. Palvelun käyttämiseksi on noudatettava Kelan ja THL:n määräyksiä, jotka koskevat muun muassa dokumenttien muotoa, sisältöä ja niiden toimittamista. Potilastiedon arkistoon määrittelyissä käytetään HL7 [24] määräyksiä ja niiden kansallisia laajennoksia.

HL7 eli *Health Level 7* on joukko kansainvälisiä määräyksiä, joilla voidaan siirtää terveydenhoidollista ja hallinnollista tietoa eri palveluntarjoajien välillä [25]. HL7 määrittelee useita standardeja, joista vanhojen potilastietojen siirtoon liittyvät ovat *HL7v3 messaging* eli sanomarajapinnan määräykset sekä CDA [26] (*Clinical Document Architecture*) eli dokumenttien rakenteiden määräykset [27]. Arkiston dokumentit noudattavat CDA R2 standardin määräyksiä kansallisilla laajennoksilla laajennettuna. Lisäksi tietyillä asiakirjojen sisällöillä on erilliset määrittelyt, mutta tässä työssä niitä ei käsitellä, sillä ne koskevat ainoastaan uusia asiakirjoja.

Potilastiedon arkistoon tallennettavat asiakirjat ovat CDA muodossa. CDA on osa HL7 standardiperhettä ja se määrittelee XML pohjaisten dokumenttien koodauksen, muodon ja semantiikan. Suomessa on käytössä kansallisia laajennoksia kansainvälisen CDA määräyksen rinnalla ja seuraavat kappaleet käsittelevät CDA R2 määritystä Potilastiedon arkiston kannalta kansallisine lisäyksineen Suomen näkökulmasta. Lähteenä on käytetty Kelan julkaisemaa määritystä [28]. Tarkoituksena on antaa riittävä kuvaus käsitteistä tulevia lukuja varten.

CDA asiakirja on XML pohjainen dokumentti, joka koostuu otsikkotiedoista ja varsinaisesta sisällöstä. Kuva 3 esittää asiakirjan loogista rakennetta. Otsikkotietoja käytetään esimerkiksi asiakirjojen etsimiseen, arkistointiin, käyttöoikeuksien hallintaan, säilytykseen ja tuhoamiseen sekä sähköisten allekirjoitusten säilytykseen. Lisäksi otsikkotiedot voivat sisältää kyseisen dokumentin tuottaneen toimittajan tietoja. Suomessa otsikkotietoja käytetään lisäksi asiakirjojen luokitteluun näkymien avulla.

Otsikkotiedot Kirjaaja Potilaan tiedot Rekisterinpitäjä Organisaatio Luontiaika							
Näkymä <table border="1"> <tr> <td>Vaihe</td> <td> <table border="1"> <tr> <td>Otsikko</td> <td> <table border="1"> <tr> <td>Merkintä</td> </tr> </table> </td> </tr> <tr> <td>PDF/teksti</td> <td></td> </tr> </table> </td> </tr> </table>	Vaihe	<table border="1"> <tr> <td>Otsikko</td> <td> <table border="1"> <tr> <td>Merkintä</td> </tr> </table> </td> </tr> <tr> <td>PDF/teksti</td> <td></td> </tr> </table>	Otsikko	<table border="1"> <tr> <td>Merkintä</td> </tr> </table>	Merkintä	PDF/teksti	
Vaihe	<table border="1"> <tr> <td>Otsikko</td> <td> <table border="1"> <tr> <td>Merkintä</td> </tr> </table> </td> </tr> <tr> <td>PDF/teksti</td> <td></td> </tr> </table>	Otsikko	<table border="1"> <tr> <td>Merkintä</td> </tr> </table>	Merkintä	PDF/teksti		
Otsikko	<table border="1"> <tr> <td>Merkintä</td> </tr> </table>	Merkintä					
Merkintä							
PDF/teksti							

Kuva 3. Asiakirjan rakenne

Vanhojen potilastietojen arkistoinnissa asiakirjat ovat joko palvelutapahtuma- tai hoitoasiakirjoja. Palvelutapahtuma kokoaa yhden hoitotapahtuman kirjaukset samaan kokonaisuuteen. Kokonaisuus voi olla esimerkiksi yksi avohoitokäynti tai vuodeosastohoitajakso. [27] Hoitoasiakirja sisältää varsinaisen hoitotapahtuman tiedon, kuten esimerkiksi lääkärin kirjaaman kertomuksen tai mittaustuloksen.

Koodin tiedot

Luokituksen nimi: AR/YDIN - Näkymät
 Version nimi: AR/YDIN - Näkymät
 Tunniste: 10
 Lyhenne: SIS
 Lyhyt nimi: Sisätaudit
 Pitkä nimi: Sisätaudit
 Määritelmä: sisätautien erikoisalaan kuuluvien potilaskertomustietojen kirjaamisessa käytettävä näkymä
 Asiakirjatyyppi: 1 Kertomusteksti
 Beskrivning: vy som används för journalanteckningar inom specialiteten inre medicin
 Erillinen asiakirja: ei
 Förkortning: MED
 Järjestys: 1
 Lisänäkymä: T
 Långt_namn: Inre medicin
 Näkymätyyppi: 1 Kertomusnäkymä
 Päänäkymä: T
 RGB: 247-127-0
 Säilytysaikaluokka: 2
 Tehtävaluokka: 06.03

Kuva 4. Esimerkki näkymästä koodistopalvelusta. [29]

Hoitoasiakirja kuuluu aina yhteen tai useampaan *näkymään*, joka kuvaa sen sisältöä. Niiden avulla voidaan myös toteuttaa pääsyoikeushallinta, jonka avulla pääsyä arkaluontoiseen tietoon, kuten psykiatriin kertomuksiin, voidaan rajoittaa. Lisäksi luokittelu auttaa ryhmittelemään tietosisältöjä eri erikoissammattiryhmiin kuten hammashoitoon. Näkymät määritellään koodistoilla ja julkaistaan kansallisessa koodistopalvelussa.

Kuva 4 näyttää SIS-näkymän sisällön koodistopalvelusta. Kuvan näkymä voi esiintyä pää- ja lisänäkymänä. Osa näkymistä voi olla vain päänäkymänä ja osa sivunäkymänä. Tieto voi kuulua useampaan näkymään, sivunäkymänä voi esimerkiksi olla *TYÖ*, mikä kertoo, että tieto liittyy työterveydenhoitoon ja päänäkymänä *S/S*, mikä tarkoittaa, että sisältö liittyy sisätauteihin. Kansallisten näkymien lisäksi voidaan käyttää paikallisia näkymiä, jotka erotetaan kansallisista näkymistä organisaatiokohtaisilla koodistoilla.

Hoitoasiakirjan sisältö voi olla sisällöltään joko tavallinen potilastiedon arkiston asiakirja tai siinä voi olla *vanhojen potilastietojen laajennus*. Tavallinen Potilastiedon arkiston kertomusasiakirja koostuu yhdestä tai useammasta merkinnästä, mikä tarkoittaa terveydenhuollon ammattihenkilön suorittamaa kirjausta, kuten esimerkiksi yksittäistä mittaustulosta tai vastaanotolla kirjattua hoitokertomusta.

Merkinnän sisältöä kuvataan kuvan 3 mukaisesti hoitoprosessin *vaiheilla* ja *otsikoilla*, jotka on kuvattu kansallisissa koodistoissa. Merkintä koostuu hoitoprosessin vaiheista, joita voivat olla tulotilanne, hoidon suunnittelu, toteutus tai arviointi. Vaihe määrittelee

hoidon kulkua potilaan näkökulmasta. Hoitoprosessin alle tulee otsikko, joka kuvaa tarkemmin kirjauksen tarkoitusta ja jäsentää kirjauksen sisältöä. Otsikoita voivat olla esimerkiksi seuraavat: Diagnoosi, esitiedot (anamneesi), ja hoidon tarve [29].

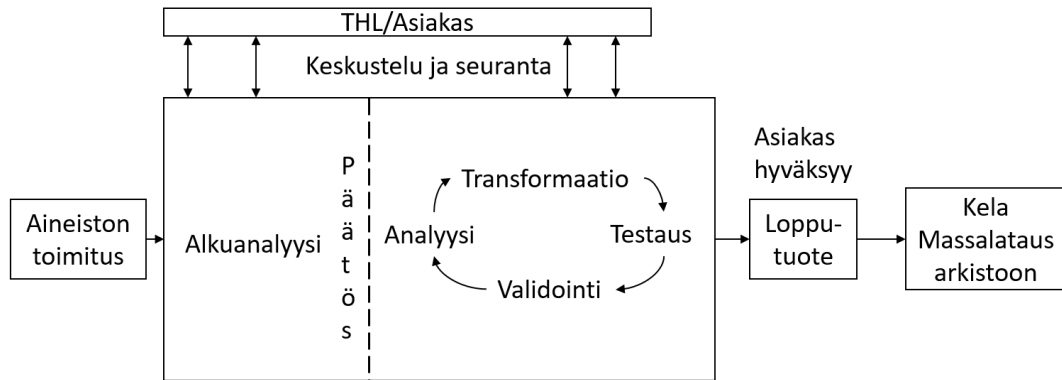
Merkinnän varsinainen sisältö koostuu *näyttömuotoisesta tekstistä* ja valinnaisesta *rakenteisesta sisällöstä*. Näyttömuotoinen teksti on tarkoitettu rakenteisen tiedon keskeisen sisällön esittämiseksi ihmisen luettavassa muodossa. Sille on määritetty tyylitiedosto, jonka avulla esimerkiksi selain osaa näyttää tekstin oikein. Rakenteinen sisältö mahdollistaa tiedon esittämisen koneluettavassa muodossa, jolloin tiedon siirto ja sen koneellinen sisällön ymmärtäminen, esimerkiksi päätöksen tuen taustalla, tulee helpommaksi. Vanhojen potilastietojen arkistoinnissa ei käytetä rakenteisia tietoja.

Vanhojen potilastietojen arkistoinnissa noudatetaan Potilastiedon arkiston määritysten lisäksi myös erityisiä vanhojen potilastietojen arkistoinnin määritelmiä [30]. Nämä määritykset kertovat, kuinka vanhalle aineistolle muodostetaan otsikkotiedot ja kuinka erilaiset tietosisällöt sijoitetaan asiakirjaan. Näitä määritelmiä tarvitaan, sillä vanhan aineiston asiakirjojen tulee noudattaa uusien arkiston asiakirjojen määritelmiä, joita ei ollut olemassa vanhojen tietojen luontihetkellä. Näiden määritysten avulla vanhoille asiakirjoille luodaan keinotekoisesti näkymä, vaiheet ja otsikot sekä palvelutapahtuma.

Vanhojen potilastietojen arkistoinnissa voidaan käyttää erikoistapausta, jossa asiakirjan sisältö voi olla myös PDF, tekstiä tai XHTML-muotoiltua sisältöä. PDF-sisällöllä voidaan tallentaa esimerkiksi skannattu lomake, jonka alkuperäinen versio on paperinen.

3.2 Vanhojen potilastietojen arkistointi -prosessi

Tässä aliluvussa käydään läpi nykyinen vanhojen potilastietojen arkistointi -prosessi, jota käytettiin analysoitavissa projekteissa. Prosessi on koostettu luvussa 4 esitellyistä projekteista tehtyjen havaintojen perusteella.



Kuva 5. Potilastietojen arkistointi -prosessi.

Potilastietojen arkistointi koostuu useasta eri vaiheesta, joita on havainnollistettu kuvassa 5. Arkistoiija suorittaa potilastietojen transformaation potilastiedon arkiston vaatimaan muotoon ja vastaa koko esitetystä prosessista. Ensimmäisessä vaiheessa asiakas toimittaa aineiston *arkistoijalle alkuanalyysiin*. Alkuanalyysin suorittamiseksi pystytetään ympäristö, jossa aineistoa voidaan analysoida. Alkuanalyysin tulosten perusteella muodostetaan arvio aineiston arkistoinnin kestosta ja haasteista. Alkuanalyysin aineiston voi olla suppeampi otos koko aineistosta, mikäli se riittää analyysin suorittamiseen. Alkuanalyysin tarkoituksena on varmistaa, että projekti voi onnistua. Jatkopäätöksen jälkeen asiakas voi toimittaa lopullisen aineiston, jolla prosessi suoritetaan.

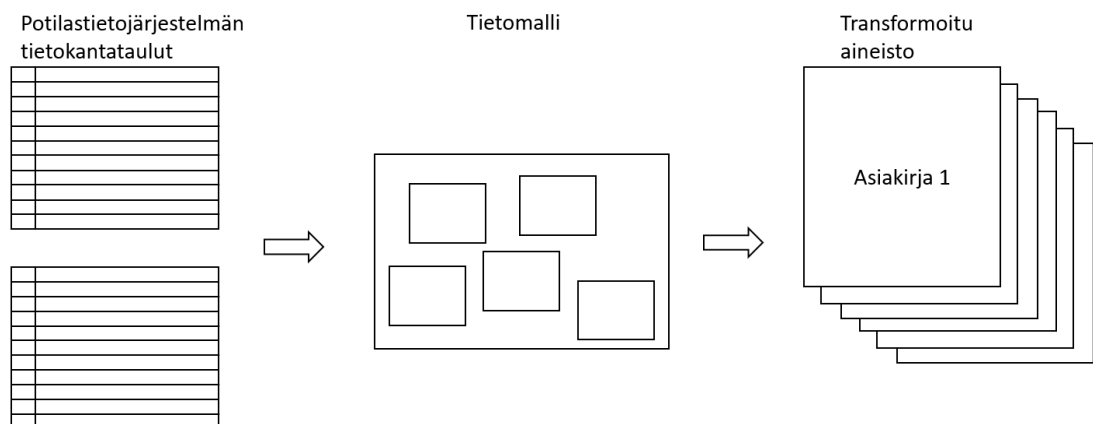
Alkuanalyysin haasteena on, että potilastiedot sijaitsevat potilastietojärjestelmän käyttämässä tietovarastossa, joka on yleensä tietokanta. Potilastietojärjestelmän tietojen säilytysmuoto on hyvin erilainen verrattuna potilastiedon arkiston arkistointimuotoon. Tavallisesti datamigraation toteuttaja tuntee lähdejärjestelmän, mutta tässä diplomityössä käsitellyissä projekteissa toteuttaja on täysin ulkopuolinen lähdejärjestelmään verrattuna. Tietojen vienti tuntemattomasta potilastietojärjestelmästä ulkopuolisena toimijana voi olla haastava käänteismallinnusprojekti, sillä alkuperäinen tietokanta on suunniteltu toimimaan potilastietojärjestelmän tietovarastona, joten sen rakenteen kartoitus ja ymmärtäminen voi olla haastavaa.

Mikäli alkuanalyysin perusteella todetaan, että projekti voidaan tehdä, aloitetaan prosessi, jossa aineistoa analysoidaan, kehitetään transformaatioita, testataan ja validoidaan. Prosessi voidaan suorittaa vaiheittain siten, että analysoidaan pieniä osakokonaisuuksia, joille kehitetään transformaatiot. Näitä pieniä osakokonaisuuksia voidaan testata ja validoida muista riippumatta. Transformaatiossa tehtyjen havaintojen perusteella

voidaan havaita tarve tarkemmalle analyysille, jolloin kehässä palataan analyysivaiheeseen. Validointivaiheessa asiakas validoi transformaatio tuloksen ja päättää sen hyväksynnästä.

Kun koko aineistolle on kehitetty transformaatiot, koko aineisto käsitellään, jolloin muodostuu prosessin lopputuote. Asiakas validoi lopputuotteen ja joko hyväksyy tai hylkää sen. Lopputuotteen hyväksynnän jälkeen aineisto siirretään Kelaan ladattavaksi. Prosessissa on mukana myös THL ja Kela, joilta voidaan pyytää ratkaisuja ongelmatilanteisiin. THL vastaa itse tietosisällön ominaisuuksista ja Kela latauksen teknisestä toteutuksesta.

Potilastiedon arkistoon vietävät tiedot ovat asiakirjoja, jotka muodostetaan koostamalla potilastietoja. Kuva 6 esittää prosessia, jossa tietokannan tauluista poimitaan tiedot tietomalliin, jonka avulla tiedoista muodostetaan asiakirjoja. Asiakirjamuodossa tiedossa ei ole enää rakennetta, eikä se todennäköisesti muistuta enää lainkaan alkuperäistä muotoa. Tietoa on koostettu ja poimittu useasta lähteestä yhteen potilasasiakirjaan.



Kuva 6. Muunnosprosessi.

Analysoinnin ja muunnoksen lisäksi prosessissa *testataan* ja *validoidaan*. Testauksessa varmistetaan, että muunnos toimii sille asetettujen vaatimusten mukaisesti ja validoinnissa varmistetaan, että vaatimukset ovat oikeat. Prosessin aikana määritykset voivat muuttua validoinnin seurauksena, jolloin testausta ja kehitystä suoritetaan lisää.

Toteutuksen ja testaamisen jälkeen asiakas validoi lopputuloksen ja hyväksynnän jälkeen aineisto viedään potilastiedon arkistoon. Tässä vaiheessa tuotetut asiakirjat siirretään Kelaan, jossa ne ladataan Potilastiedon arkistoon. Kun Kela on ladannut aineiston, se tuottaa lataus- ja virheraportin, josta voidaan todeta onnistuminen tai mahdolliset vir-

heelliset asiakirjat. Virheelliset asiakirjat korjataan ja asiakirjat toimitetaan uudelleen. Mikäli myöhemmin huomataan, että asiakirjat olivat virheellisiä, ne korvataan rajapintaa käyttämällä. Virheettömän latausraportin jälkeen alkuperäinen aineisto voidaan poistaa.

Vanhojen tietojen arkistointi edellyttää liittymistä Kanta-palveluihin. Liittymisen jälkeen Kelaan toimitetaan testimateriaalia, jolla varmistetaan, että kyetään tuottamaan Kelan teknisesti hyväksymää aineistoa. Arkistojen tehtävänä on varmistua siitä, että potilastiedot ovat säilyneet muuttumattomina ja kaikki tiedot tulevat arkistoiduksi.

3.2.1 Aineiston toimitus

Aineistotoimitukset voidaan jakaa kahteen ryhmään sen perusteella, onko niille suoritettu aineistopoiminta. Nämä ryhmät ovat *valmiiksi poimittu* ja *poimimaton aineisto*.

Valmiiksi poimittu aineisto tarkoittaa aliluvun 2.1.1 mukaan lähdejärjestelmästä luettua tietoa, joka on sovitettu uuteen tietomalliin. Tällainen on esimerkiksi lähdejärjestelmän tietokannasta poimittu aineisto, joka sisältää vain potilastietoja sopivassa tietomallissa. Toiseen ryhmään kuuluvat aineistot, jotka pitää poimia ennen tarkempaa käsittelyä. Tällaisia aineistoja ovat, esimerkiksi kokonainen lähdejärjestelmän tietokanta tai kopio lähdejärjestelmästä. Mikäli lähtöaineisto on poimittu aineisto, voidaan muunnos Kanta-arkiston vaatimaan muotoon suorittaa helpommin, kuin tilanteessa, jossa lähtöaineistona on tuntemattoman potilastietojärjestelmän tietokanta, eikä testaukseen ole käytettävissä vastaavaa potilastietojärjestelmää.

Valmis poiminta on arkistojen kannalta helpoin tilanne. Se voidaan toteuttaa siten, että asiakas tilaa lähdejärjestelmän toimittajalta tietojen poiminnan tietokannasta. Tietokannasta suoritettussa poiminnassa tietokannasta on eroteltu oleellinen aineisto ja se on muunneltu siten, että se on dokumentoidussa helposti käsiteltävässä muodossa. Tässä tilanteessa arkistojen vastuulle jää vain aineiston muuntaminen oikeaan muotoon sen sijaan, että analysoitaisiin alkuperäinen järjestelmä siirrettävän tiedon kartoittamiseksi. Vastuu tiedon oikeellisuudesta ja kattavuudesta on silloin asiakkaalla.

Poimimaton aineisto voidaan toimittaa esimerkiksi *kokonaisena järjestelmänä*. Kokonaisuudessa järjestelmässä tiedot ovat siinä muodossa, missä lähdejärjestelmä niitä käsittelee, eli raakamuodossa. Kokonainen järjestelmä voi koostua pelkästään tietokannasta tai koko potilastietojärjestelmästä ohjelmistoineen. Aineiston sijaitessa esimerkiksi kolmannen osapuolen järjestelmässä, asiakas ei voi itse suorittaa aineiston noutoa, vaan se on tilattava järjestelmän ylläpitäjältä.

Kokonainen järjestelmä voidaan toimittaa esimerkiksi *tietokantavedoksena*. Tietokantavedoksen toimittaminen on asiakkaalle yksinkertainen vaihtoehto. Tavallisesti tietokanta

on jakautunut useaan tiedostoon ja useaan eri paikkaan, eikä sen siirtäminen ole mahdollista. Tietokantavedos sisältää tietokannan sisällön ja rakenteen yleensä yhtenä tiedostona ja vedos voidaan toimittaa eteenpäin. Toimittamisen jälkeen arkistoiija palauttaa vedoksesta *tietokantakopion* eli alkuperäistä vastaavan tietokannan, jolla seuraavat vaiheet (analyysi ja varsinainen muunnos) voidaan suorittaa. Asiakkaalle jää silloin vastuu vedoksen sisällöstä ja oikeellisuudesta, sillä arkistoiija ei pääse lainkaan käsiksi alkuperäiseen järjestelmään. Kun aineisto toimitetaan tietokantavedoksena, arkistoiija pääsee käsiksi järjestelmän tietoon, muttei sovellukseen, eli arkistoiijan on tehtävä enemmän työtä järjestelmän toiminnan ymmärtämiseksi, sillä hän ei pääse tarkastelemaan miltä asiat näyttivät alkuperäisessä järjestelmässä.

Sen sijaan luovutettaessa järjestelmä tietokantoiheen ja ohjelmistoiheen, arkistoiija näkee täyden kokonaisuuden alkuperäisestä järjestelmästä ja voi myös päästä kokeilemaan järjestelmää, mikä voi auttaa projektin onnistumista. Jos kokonaisen järjestelmän toimittaminen ei ole mahdollista, pääsy alkuperäiseen järjestelmään voidaan mahdollistaa, tietosuoja-vaatimukset huomioiden, esimerkiksi luomalla tunnukset, joilla arkistoiija pääsee järjestelmään käsiksi. Lisäksi toimitetaan erillinen tietokantavedos, jotta arkistoiija saa pääsyn sekä sovelluksen tietoon, että itse sovellukseen.

On syytä huomioida, että työssä käsitellyt järjestelmät olivat siirron aikana pois käytöstä, jolloin niihin ei syntynyt uutta sisältöä siirron aikana. Yleisestikään järjestelmiin ei synny uutta sisältöä, sillä Kela on mahdollistanut sähköisen arkistoinnin vasta vuoden 2014 jälkeen, joten vanhat järjestelmät ovat yleensä olleet poissa käytöstä jo kauan. Tästä seuraa se, että aineisto voidaan toimittaa projektin alussa yhdessä osassa, eikä lopussa tarvita uutta täydennystä. Jos järjestelmä olisi käytössä siirron aikana, se pitäisi huomioida erillisellä loppuarkistoinnilla tai käyttökatkolla. Edellisestä johtuen järjestelmä saattaa kuitenkin olla ollut käyttämättömänä jo hyvin kauan, jolloin järjestelmän alkuperäiset asiantuntijat eivät mahdollisesti ole enää käytettävissä.

3.2.2 Analyysiympäristön pystyttäminen

Aineiston vastaanottamisen jälkeen on pystytettävä ympäristö, jossa sitä voidaan käsitellä. Aineiston tietokantavedoksesta voidaan muodostaa lähdejärjestelmän tietokantaa vastaava tietokantakopio. Jos aineistona saatiin virtuaaliympäristö, on se otettava käyttöön. Analyysi voidaan myös suorittaa asiakkaan ympäristössä, johon luodaan yhteydet tarpeen mukaan. Jos aineistona on asiakkaan järjestelmästä valmiiksi poimittu otos, ympäristön pystytystä ei välttämättä vaadita lainkaan.

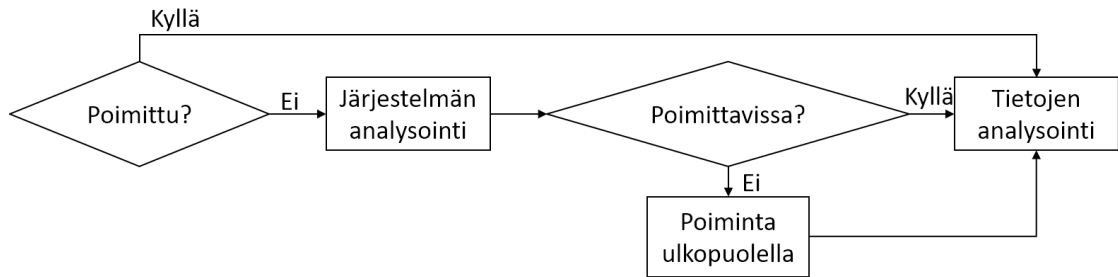
Tietokantavedoksesta muodostettava tietokanta voi vaatia useita työvaiheita. Erilaisia tietokannan hallintajärjestelmiä on useita, joten on varauduttava siihen, että tulevaisuudessa tietokantakopion pystyttäminen vedoksesta voi olla hyvin haastavaa, sillä samaa tietokantajärjestelmää ei ole aina helposti saatavilla. Saatavuuden lisäksi ongelmaksi voi muodostua myös eri tietokantajärjestelmien lisenssien kustannukset. Ongelmia saattaa aiheutua myös vanhan tietokannan käytöstä nykyaikaisessa ympäristössä. Esimerkiksi Windows Server 2000 [31] ympäristössä toimiva ohjelma, voi olla hyvin hankala saada toimimaan nykyaikaisessa Windows-ympäristössä.

Kun käytössä on alkuperäinen järjestelmä, tai sen kopio, ympäristön pystyttäminen on suhteellisen yksinkertaista. Riittää, että jokaiselle projektin jäsenelle annetaan joko kopio tai pääsy järjestelmään. Pääsy voidaan järjestää esimerkiksi etäyhteydellä tai virtuaaliympäristön kopiolla.

Aliluvussa 3.1.1 selostetun aineiston arkaluontoisuuden ja tietojenkäsittelysopimuksen vuoksi, analyysia ei ole hyvä suorittaa oikealla tunnistettavalla potilastiedolla, vaan aineisto tulisi muokata siten, että alkuperäisiä henkilöitä ei voida tunnistaa. Tämä voidaan suorittaa esimerkiksi korvaamalla jokaisen potilaan henkilötunnus ja muut henkilötiedot keksityillä vaihtoehdoilla. Kun käytössä on kopio tietokannasta tai järjestelmästä, aineisto voidaan pseudonymisoida, eli vaihtaa henkilöiden tunnistetut keinoiksi. Jos tutkiminen suoritetaan alkuperäisessä järjestelmässä, analysointiin voitaisiin käyttää sijaisjärjestelmää eli esimerkiksi asiakkaan testi- tai koulutusjärjestelmää. Sijaisjärjestelmää käytettäessä oleellista on varmistua siitä, että sen ja alkuperäisen järjestelmän toiminnallisuudet ja versiot ovat samat. Aina se ei kuitenkaan ole mahdollista, sillä sijaisjärjestelmää ei aina ole, tai sen versio on eri kuin alkuperäisen järjestelmän.

3.2.3 Arviointi ja analyysi

Ennen projektiin ryhtymistä, tehdään arviointi, jonka perusteella varmistetaan, että projekti voi onnistua. Arvioinnin perusteella asiakkaalle annetaan tarjous projektin kustannuksista ja kestosta. Tarjouksen hyväksymisen jälkeen suoritetaan vielä enemmän analysointia, jossa muun muassa luokitellaan siirrettävä tieto oikeisiin Arkiston näkymiin.



Kuva 7. Analyysiprosessi

Kuva 7 esittää alkuanalyysin kulkua. Analyysin kulku riippuu siitä, onko aineisto valmiiksi poimittu. Jos aineistoa ei ole poimittu valmiiksi, kuvan mukaisesti alkuanalyysin ensimmäisenä tarkoituksena on selvittää, voidaanko aineistopoiminta tehdä itse. Arviointi tehdään järjestelmää analysoimalla. Mikäli todetaan, että poimintaa ei voida tehdä itse, asiakkaan kanssa neuvotellaan, miten aineisto saadaan poimittua.

Jos aineisto oli valmiiksi poimittua tai selvityksessä todetaan, että aineisto voidaan poimia itse, seuraavaksi tarkoituksena on selvittää, mitä tietoja järjestelmä sisältää ja kuinka paljon työtä sen arkistointi vaatii. Tarkoituksena on saada projektille selkeät puitteet, eli arvio projektin kestosta, sekä vaadittavat henkilö- ja työresurssit. Tarkoituksena on myös muodostaa käsitys järjestelmän kompleksisuudesta ja riskeistä. Perehtymisessä voidaan käyttää apuna asiakkaan osaamista.

Alkuanalyysin jälkeen suoritetaan tarkempi analyysi, jossa tavoitteena on järjestelmän sisällön kattava kartoitus. Tarkoituksena on selvittää mitä tietoa järjestelmästä löytyy ja päättää, mitä löydetylle tiedolle tehdään. Löydöt dokumentoidaan, jotta lopussa voidaan varmistaa, että kaikki tiedot todella käsiteltiin. Asiakkaan kanssa käydään lopuksi läpi, mitä saatujen tietojen perusteella arkistoidaan ja mitä ei voida arkistoida.

Kaikkea potilastietoa ei voida viedä arkistoon. Arkisto ei tue esimerkiksi videoiden tai äänen tallentamista, joten nämä muodot on tallennettava ja säilytettävä toistaiseksi säilytysaikojen mukaisesti. Myöhemmin käyttöön otettava kansallinen kuvantamisen arkisto toteutetaan kuvantamisen arkistoinniksi [32]. Muita esimerkkejä asiakkaalle jäävistä tiedoista ovat käyttölokit ja tilastointitiedot.

Luvussa 2.1.1 mainittu aineiston profilointityökalu voi tarkoittaa esimerkiksi tietokantakyselyiden käyttämistä aineiston laadun selvittämiseen [9]. Yksinkertaisilla kyselyillä voidaan varmistaa, että tietokannan taulussa ei ole orporivejä tai tyhjiä arvoja kohdissa, joissa niitä ei saisi olla. Samalla voidaan etsiä selkeitä testipotilaita, kuten potilaita, joiden nimi on Aku Ankka. Aineistosta saattaa kuitenkin löytyä epäilyttävän nimisiä henkilöitä,

sillä kirjausvaiheessa ei ole aina tiedossa potilaan nimeä, jos potilas on esimerkiksi tajuton. Myös henkilötunnus saattaa olla vääränlainen, jos henkilö on esimerkiksi ulkomaalainen. Kun tällaiset tiedot selvitetään alkuvaiheessa, transformaatiovaihe on helpompi.

3.2.4 Transformaatio

Analysoinnin jälkeen arkistoitava tieto transformoidaan arkistoitavaan muotoon. Transformaatiossa järjestelmän tiedoista muodostetaan asiakirjoja, jotka esittävät poimitun aineiston oleelliset osat.

Transformaatiossa mietitään, miten merkintöjä koostetaan. Yhdelle monisairaalle potilaalle on saattanut kertyä esimerkiksi useita kymmeniä laboratorionäytteitä, jolloin kaikkien merkintöjen kirjaaminen omaan asiakirjaan vaikeuttaisi tietojen käsittelyä hoitotilanteessa. Sen sijaan yhteen asiakirjaan voidaan koostaa useampia merkintöjä. Esimerkiksi laboratorionäytteiden mittaustulokset voidaan koostaa yhteen asiakirjaan tutkimuksen mukaan tai mittaustulokset voidaan yhdistää lähetteen perusteella. Vuodeosaston seurantakirjauksia voidaan koostaa esimerkiksi hoitojaksojen mukaan tai kuukausittain. Kaikki rokotusmerkinnät voidaan koostaa yhteen asiakirjaan, jolloin niiden katselu on yksinkertaista.

Vapaamuotoinen tekstiesimerkki:

Hoidon syy

Kovat rintakivut, lähettäjä: tkl. P. Kukunor

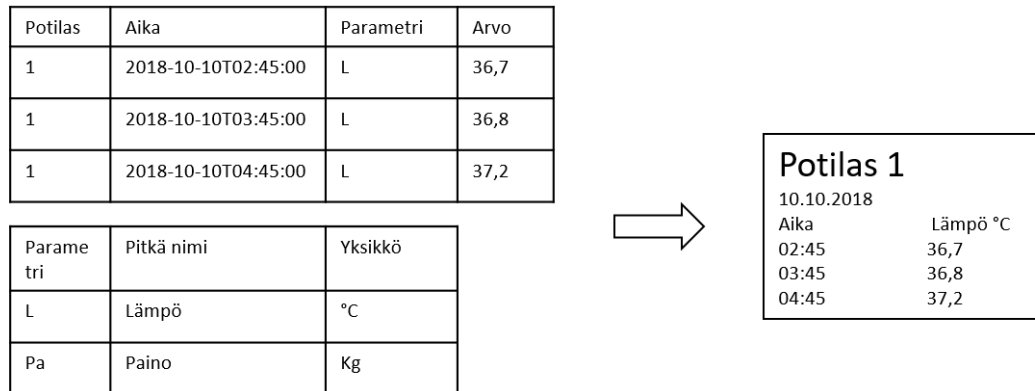
Esitiedot (anamneesi)

Sairastanut diabetesta vuodesta 1959. Nykyhoitoja Insulin lente 24 KY + 8 KY. Neuro-, nefro- ja retinopatia on todettu. Munuais insuffisienssi vaikea-asteinen, krea ollut 460 tasoa. Verisuonikomplikaationa todettu jo klaudikaatio vas. alaraajassa. Ei lääkitystä. Tupakoinut, mutta lopettanut viime syksynä.

Noin 2 kk sitten alkanut esiintyä puristavia rintakiputuntemuksia. Kipu säteilee kuristavana kaulalle. Oireet esiintyvät fyysisessä rasituksessa, kiirehtiessä, vastamäessä tai portaita noustessa. Menee pysähtyessä ohi. Kokeillut Nitroa, joka myös auttaa.

Kuva 8. Esimerkki näyttötekstimuotoisesta merkinnästä. [33]

Kuva 8 esittää näyttötekstimuotoista merkintää. Merkinnässä nähdään hoitoprosessin vaihe sekä otsikot. Tämä merkintä voisi näyttää hyvin samalta myös lähdejärjestelmässä, joten muunnos voi tapahtua kokoamalla tietokannan taulusta tekstirivit ja muodostamalla merkinnälle hoitoprosessin vaiheet ja otsikot esimerkiksi lähdejärjestelmän metatietojen avulla. Näyttömuotoisen tekstin lisäksi voidaan käyttää PDF-muotoa, tekstiä tai XHTML-muotoiltua tekstiä. PDF-muotoa voidaan käyttää esimerkiksi skannattujen paperilomakkeiden tai kuvien arkistoinniseksi.



Kuva 9. Tyypillinen muunnos

Kuva 9 esittää muunnosta, jossa lähdejärjestelmän taulut sisältävät lämpötilan mittaus-tietoja. Yksi taulu sisältää mittaustulokset ja toinen taulu sisältää mittauksessa käytetty-jen mittayksiköiden tiedot, joihin mittaustuloksissa viitataan. Muunnoksessa taulujen tie-dot yhdistetään, jolloin saadaan muodostettu yhden potilaan osalta tekstimuotoinen asia-kirjamainen esitys lämpötilan kehityksestä. Tiedosta on hävinnyt kaikki rakenteisuus. Hy-vin tehdyn taustatyön, eli alkuanalyysin, jälkeen muunnos on selkeä ja helppo toteuttaa.

3.2.5 Testaus

Aineiston itsenäiseen tekniseen validointiin voidaan käyttää Kelan validointityökalua [34], joka varmistaa aineiston teknisen oikeellisuuden. Testiaineistoon poimitaan mahdollisim-man laajasti erilaista sisältöä, jotta validoinnista tulisi mahdollisimman kattava. Validointi on kuitenkin vain tekninen eikä ota kantaa aineiston sisältöön.

Kehityksen aikana ohjelmiston toimintaa testataan kattavasti. Analyysistä muodostetut oletukset järjestelmästä koodataan ohjelmistoon siten, että oletuksia rikkovat tilanteet aiheuttavat virheen. Virhetilanteissa oletuksia tarkastetaan ja tarpeen vaatiessa niitä muutetaan. Ohjelmistoon voidaan tehdä myös käsittelijöitä, jotka ohittavat tietynlaiset virheet, jos aineistossa on virheitä, joita ei voida korjata.

Lisäksi kehittäjä testaa ohjelmaa ajamalla potilaita prosessin läpi ja tarkistamalla, että muunnettu aineisto vastaa lähtöaineistoa. Tässä oleellisena vaiheena on valita mahdol-lisimman kattavia testitapauksia, jotta testauksen tuloksesta saadaan mahdollisimman totuudenmukainen.

Kela vaati aiemmin, että se suorittaa aineiston teknisen validoinnin ennen testauksen hyväksyntää. Validointi suoritettiin muodostetulle aineistolle eli XML-asiakirjoille Kelan

validointityökalulla. Testiaineisto ei saa sisältää aitoa potilastietoa, vaan kaikki yksilöivät tiedot on poistettava esimerkiksi korvaamalla ne pseudonymisoiduilla vastineilla. Testaamiseen olisi suositeltavaa käyttää testikäyttöön luotua testiaineistoa. Henkilötunnukset vaihdetaan Kelan toimittamiin testihenkilötunnuksiin.

3.2.6 Validointi

Validointi on asiakkaan suorittama varmistus siitä, että ohjelma toimii alkuperäisen ajatuksen mukaisesti. Validointi eroaa testauksesta siinä, että testauksessa varmistetaan, kehitetäänkö ohjelmaa oikein ja validoinnissa varmistetaan, että kehitetäänkö oikeaa ohjelmaa. Kela vaati ennen, että heidän on suoritettava aineiston tekninen hyväksyntä, mutta nykyään arkistoija vastaa itse aineiston teknisestä validoinnista ja asiakas suorittaa aineiston loppuhyväksynnän.

Migraation validointi voi lisätä suuresti prosessin kestoa, hintaa ja monimutkaisuutta. Koko aineiston validointi ei ole mahdollista vaan tavallisesti valitaan pieni osajoukko, jolla migraation onnistuminen validoidaan. Perinteisesti validoitavan osajoukon koko on valittu subjektiivisesti ja se on voinut olla esimerkiksi 15 prosenttia. Terveystietojärjestelmien migraatioiden validointiin on kehitetty erilaisia validointimenetelmiä, kuten esimerkiksi näytteistäminen. [35]

Kehityksen aikana kehittäjille jää paljon vapauksia muun muassa esitysmuodon valinnan suhteen, eikä ole järkevää kysyä asiakkaalta jatkuvasti kysymyksiä, sillä se aiheuttaisi tarpeettomia viivästyksiä kehitykseen. Sen sijaan suoritetaan testauskierroksia, joissa lähtöaineistosta valitaan sopiva määrä potilaita, jotka asiakas käy läpi ja tarkistaa, että muunnettu aineisto ja lähtöaineisto vastaavat toisiaan. Samalla voidaan miettiä, että onko aineisto sellaista, josta on myöhemmin hyötyä hoidossa, vai joudutaanko esimerkiksi aineiston esitysmuotoa vielä muuttamaan. Testauskierroksia voidaan suorittaa esimerkiksi aina yhden tietotyypin kehityksen valmistumisen jälkeen, jotta palautteeseen voidaan reagoida mahdollisimman nopeasti.

Validointi olisi hyvä suorittaa testaukseen kehitetyllä testiaineistolla, mutta se ei välttämättä vastaa oikeaa käytössä olevaa aineistoa, tai sellaista ei ole lähdejärjestelmässä saatavilla. Tällöin testaus on suoritettava oikealla potilastiedolla, joten testiympäristö on silloin suojattava yhtä hyvin kuin tavalliset potilastietojärjestelmät.

3.2.7 Tuotantoonvienti

Testauksen ja validoinnin hyväksymisen jälkeen aineisto on valmis tuotantoonvientiin. Vanhoja tietoja ei voi lähettää potilastiedon arkistoon itse, vaan on käytettävä Kelan massalatausta. Massalatauksessa Kelaan toimitetaan aineisto, jonka Kela lataa potilastiedon arkistoon eräajona. Aineisto voidaan toimittaa vaihtoehtoisesti joko levylaitteella tai *SFTP-palvelimella*.

Levylaitetta käytettäessä Kela lähettää arkistojalle levylaitteen, johon valmis aineisto ladataan. Levylaite lähetetään sen jälkeen takaisin Kelaan. Levylaitteella tiedot ovat salatuina ja salauksen purkava väline ei kulje levylaitteen mukana. SFTP-palvelinta [36] käytettäessä arkistoija pystyttää palvelimen, johon valmis aineisto ladataan. Kela lataa aineiston palvelimelta ja siirtää sen potilastiedon arkistoon.

Latauksen jälkeen asiakas suorittaa vielä muutaman potilaan kattavan pistokokeen, jolla varmistetaan, että aineisto on ladattu onnistuneesti. Luvussa 4 esitellyissä projekteissa pistokokeet ovat onnistuneet, mutta virhetilanteessa vanha aineisto pitäisi mitätöidä virheellisten asiakirjojen osalta, ja niiden tilalle toimitettaisiin uudelleen versioidut asiakirjat esimerkiksi palvelurajapinnan avulla.

4. PROJEKTIENTEN ESITTELY

Työssä analysoitiin neljä projektia, joissa tehtiin potilastietojen arkistointi. Projekteissa A ja B aineistotyyppinä oli poimimaton aineisto ja projekteissa C ja D poimittu aineisto. Projektien tavoitteina oli siirtää asiakkaiden vanhojen potilastietojärjestelmien potilastiedot potilastiedon arkistoon, jolloin vanhoista potilastietojärjestelmistä päästiin luopumaan. Järjestelmistä luopumisen lisäksi arkistointi mahdollistaa vanhojen potilastietojen katselun myös muista Kanta-yhteensopivista järjestelmistä.

Ensimmäisen projektin (A) tavoitteena oli hankkia kokemusta potilastietojen arkistoinnista. Projektin aikana arkistojia kehitti ohjelmistoratkaisun [5], jota käytettiin muissakin arkistointiprojekteissa. Lopuissa projekteissa kehitettiin vain ohjelmistoon liitettävät komponentit, joilla projektin potilastietojärjestelmän tiedot luettiin. Ohjelmistoratkaisu muodosti niistä asiakirjoja, jotka arkistoitiin.

A projektin kokemusten perusteella seuraavan projektin (B) oletettiin sujuvan yhtä hyvin, mutta projekti oli hyvin haasteellinen ja aikataulu venyi suuresti. Suurin ero näiden kahden projektin välillä oli, että projektin B potilastietojärjestelmä oli ominaisuuksiltaan todella paljon laajempi ja käsiteltävien tietotyyppien määrä oli paljon suurempi, vaikka projektin A järjestelmässä oli potilaita huomattavasti enemmän.

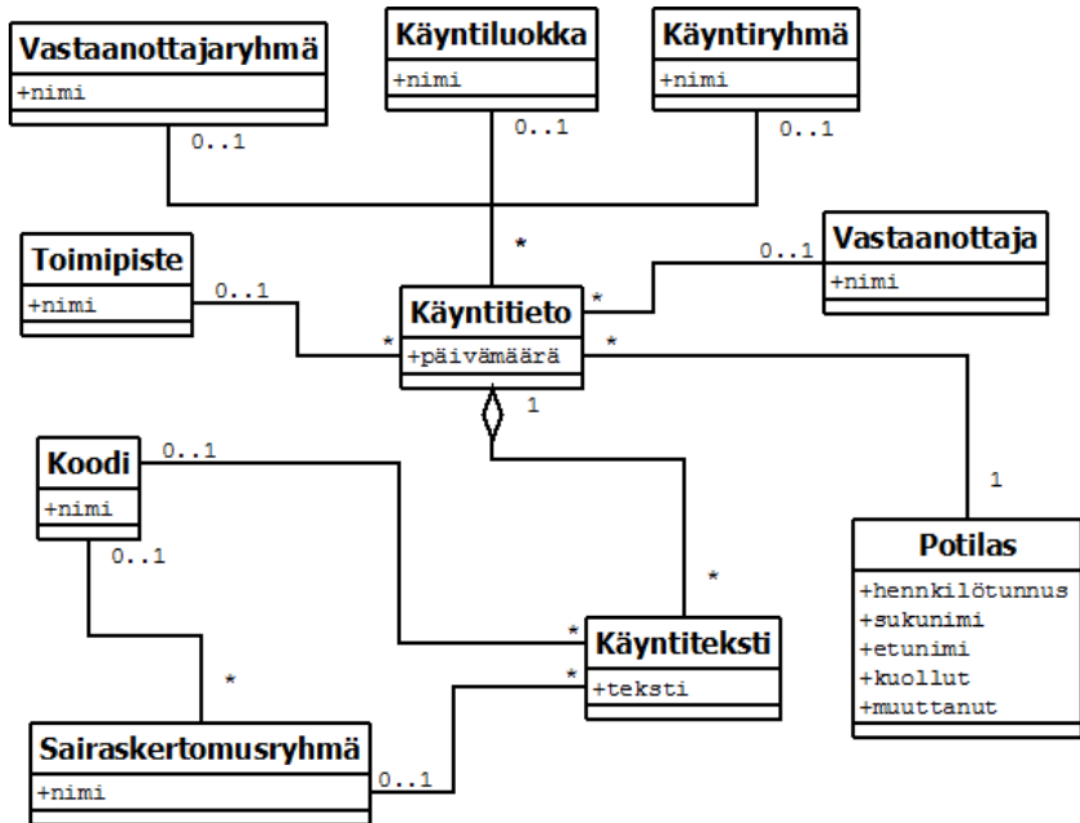
Kaksi seuraavaa projektia (C ja D) perustuivat samaan potilastietojärjestelmään. Järjestelmän toimittaja tarjoaa aineiston poimintapalvelua, jota projekteissa käytettiin. Asiakas sai palvelun avulla järjestelmästä poimitun aineiston ja toimitti sen arkistojalle muunnettavaksi. Aineisto toimitettiin CSV-tiedostoina ja siihen sisältyi ohjeet aineiston tulkinnalle. Valmiista aineistopoiminnasta oli suuri apu projektin läpiviennille.

4.1 Projekti A (Poimimaton aineisto)

Projektissa A potilastietojen arkistointi tehtiin järjestelmään, joka oli potilaiden kokonaismäärältään suuri, potilaita järjestelmässä oli noin 135 000, mutta järjestelmän tietojen sisällön ja toiminnallisuuden osalta järjestelmä oli melko yksinkertainen. Järjestelmä oli poistettu kirjauskäytöstä, mutta se oli edelleen katselukäytössä. Projektin lähtöaineistoksi saatiin tietokantavedos potilastietojärjestelmän Microsoft SQL Server 2000 [37] tietokannasta. Projektissa ei ollut lainkaan pääsyä alkuperäiseen järjestelmään.

Kuva 10 esittää järjestelmän tietomallia potilastietojen osalta, joka muodostettiin tietokannan analysoinnin perusteella. Tietomallista nähdään, että potilaskohtaiset merkinnät

koostuivat käyntitiedoista, jotka koostuivat käyntiteksteistä. Tavallisen tekstin lisäksi, näihin kenttiin oli tallennettu myös esimerkiksi kuulomittausten kuulokäyrät merkkipohjaisella grafiikalla.



Kuva 10. Projektin A järjestelmän yksinkertaistettu tietomalli. [5]

Projekti A oli ensimmäinen potilastietojärjestelmien arkistointiprosessi, mikä aiheutti omat haasteensa projektin läpiviennille. Kirjoittaja oli tässä projektissa mukana vain loppussa, joten ongelmia oli todennäköisesti enemmän, kuin tässä kerrotut.

4.2 Projekti B (Poimimaton aineisto)

Projektissa B potilastietojen arkistointi tehtiin lukukäytössä olevalle järjestelmälle, jossa potilaita oli noin 28 000, mutta järjestelmässä oli huomattavasti enemmän ominaisuuksia kuin projektin A järjestelmässä. Lähtöaineistoksi saatiin virtuaaliympäristö, joka sisälsi potilastietojärjestelmän tietokannan sekä asiakasohjelman. Käytössä oli siis varsinainen järjestelmä, jota myös loppukäyttäjät olivat käyttäneet, mikä helpotti suuresti projektin läpivientä.

Lähdejärjestelmän arkistoitavat tiedot olivat lääkäreiden kirjaukset vastaanotoilla, laboratorionäytteet ja niiden lausunnot, erilaiset mittaustulokset, vuodeosaston kirjaukset ja mitaukset, rokotukset, työterveyshuollossa syntyneet kirjaukset, lääkkeiden määräykset

sekä erilaiset lomakkeet. Järjestelmän ominaisuuksien määrä kasvatti arvioitua työmäärää, sillä järjestelmässä oli paljon erilaisia tietomuotoja, eikä tietotyyppien transformaatiototeutuksia voitu hyödyntää muissa tietotyypeissä.

Suurin haaste oli järjestelmän käyttämän lomakemuodon purkaminen. Lähdejärjestelmä käytti omaa tiedostomuotoa, jolla lomakkeiden rakenne tallennettiin tietokantaan. Kun käyttäjä avasi lomakkeen kirjattavaksi potilaalle, järjestelmä luki lomakkeen rakenteen tietokannasta ja avasi lomakkeen käyttäjälle muokattavaksi. Lomakkeelle syötetyt tiedot tallentuivat omaan tietokantatauluun. Kun lomake myöhemmin avattiin, järjestelmä haki lomakkeen rakenteen ja syötetyt tiedot tietokannasta ja muodosti näiden avulla lomakkeen näkymän.

Lähdejärjestelmä oli ollut käytössä kuntayhtymässä, mutta kuntayhtymä oli purettu käytön aikana. Arkistointi toteutettiin lakkautetun kuntayhtymän yhdelle kunnalle ja haasteena oli erottaa tähän kuntaan kuuluvat kirjaukset. Projektin alussa oli käsitys, että tiedot on jo poistettu, mutta projektin kuluessa huomattiin, että järjestelmästä löytyy myös muiden kuntien tietoja.

Projektissa potilastiedot jaettiin kahteen rekisteriin, joista toiseen siirrettiin julkisen terveydenhuollon potilastiedot ja toiseen työterveyden potilastiedot. Työterveystiedot syntyivät työterveyskäynneillä, jolloin ne kuuluivat omaan rekisteriinsä työnantajan mukaan lajiteltuina. Aineistosta ei kuitenkaan löytynyt aina yksiselitteistä työnantajan tunnistetta, joka arkistointiin vaadittiin. Tästä ongelmasta keskusteltiin THL:n kanssa, joka linjasi, että työterveystiedot on vietävä työterveysrekisteriin, mutta mikäli työnantajaa ei tiedetä, sen tilalla voidaan käyttää uutta koodia, joka tarkoittaa, että työnantaja ei ole tiedossa.

Tietokannasta löytyi sisältöä, jota käyttöliittymästä ei voinut nähdä lainkaan. Tämä ei ole yllätyksellistä, sillä ohjelmistot muuttuvat ja jotkut tietokannan tauluista saattavat jäädä turhiksi. Järjestelmään oli tehty päivityksiä, joissa tietoja siirrettiin tietokantojen tauluista uusiin tauluihin, mutta vanhat taulut jätettiin tietokantaan tietoineen. Piilossa olevat tiedot sisälsivät kuitenkin potilastietoja, jota ei voitu hävittää ja nämä tiedot päätettiin antaa asiakkaalle säilytettäväksi, kunnes selviää, mitä tiedoille tehdään.

4.3 Projekti C (Valmis aineistopiminta)

Projektin C lähdejärjestelmä oli työterveyshuollossa käytetty potilastietojärjestelmä, joka sisälsi potilaita noin 18 000. Lähtöaineistona saatiin asiakkaan tilaama valmis aineistopiminta tietokannasta. Potilastietojärjestelmän valmistaja oli poiminut potilastiedon ja se toimitettiin CSV-muotoisina tekstitiedostoina ja erillisinä liitetiedostoina. Projektin aikana ei nähty lainkaan alkuperäistä järjestelmää.

Järjestelmä sisälsi samoja tietotyypppejä kuin B-projektin järjestelmä, mutta valmiista aineistopoiminnasta johtuen, projektissa oli paljon vähemmän haasteita. Aineistopoiminnan mukana tulleiden ohjeiden avulla, aineiston ymmärtäminen oli vaivatonta, eikä selvitystyötä jouduttu tekemään.

Projektin suurimpana haasteena olivat ulkoiset liitetiedostot. Liitetiedostot olivat kuvia, PDF-, Word-, tai RTF-dokumentteja, mutta arkistoinnissa voitiin käyttää vain PDF-dokumentteja. Projektissa kaikki liitetiedostot muunnettiin PDF-muotoon, mutta muuntaminen oli haastavampaa kuin ajateltu. Osittain tämä johtui siitä, että Kelan tarjoama validointityökalu validoi aineiston eri tavoin kuin lopullisessa latauksessa käytetty latausjärjestelmä.

Myös tässä projektissa ongelmia aiheuttivat puutteelliset työnantajamerkinnot. Työnantajalle ei löytynyt aina tunnistetta ja lisäksi aineistosta löytyi tilanteita, jossa työnantajaa ei voitu päätellä, sillä potilaalla oli useita samanaikaisia työsuhteita ja työsuhteen yhdistäminen työterveyskäyntiin tehtiin vain päivämäärien avulla.

Projekteissa A ja B testaukseen käytettiin Kelan palvelua, joka aiheutti riippuvuuden Kelan aikatauluihin. Projektin C aikana otettiin käyttöön arkistojen oma testiympäristö, jolla asiakas testasi aineistoa. Oman ympäristön käyttö nopeutti testaushavaintoihin reagoimista ja lisäsi testaukseen käytettävän ajan määrää.

4.4 Projekti D (Valmis aineistopoiminta)

Projekti D perustui samaan järjestelmään kuin projekti C ja projektin aineiston oli poiminnut järjestelmätoimittaja projektin C tapaan. Projekti oli ensimmäinen, jolla päästiin arvioimaan jo kerran arkistoidun järjestelmän arkistointia. Projektissa havaittiin, että kun aineistopoimintaa ei tee itse, on tarkkaan sovittava missä muodossa aineisto saadaan ja huolehdittava siitä, että aineisto poimitaan myös seuraavissa projekteissa samalla tavalla. Valmis aineistopoiminta aiheuttaa riippuvuuden aineistopoimijaan, mikä täytyy huomioida riskejä arvioitaessa. Poimintaongelmista huolimatta projektissa D voitiin käyttää projektin C ohjelmistokomponenttia. Projektin C ohjelmistokomponentin käyttö onnistui hyvin ja ohjelmistokomponentin jatkokehitykseen kuluva aika oli pieni projekteihin A, B ja C verrattuna.

Muita haasteita aiheuttivat salatut liitetiedostot. Järjestelmän käyttäjät olivat suojanneet salasanoilla erityisen arkaluontoisia asiakirjoja, mutta he eivät muistaneet käytettyjä salasanoja. Osa salasanoista oli kirjattu liitetiedoston selostukseen, mutta suurin osa salasanoista jouduttiin murtamaan, jotta ne saatiin avattua ja arkistoitua. Kaikki salatut tiedostot saatiin avattua.

5. VANHOJEN POTILASTIETOJEN ARKISTOINTI -PROSESSIN HAASTEET

Seuraavassa kerrotaan arkistointiprojekteihin liittyvistä ongelmista aliluvuittain prosessin vaiheiden mukaan. Projektien vaiheet eroavat toisistaan aineistotyyppin mukaan, joten myös niiden haasteet ovat erilaisia. Yleensä poimittuun aineistoon liittyy vähemmän haasteita. Eniten ongelmia tulee vastaan, kun aineistopoiminnan tekee itse täysin tuntemattomaan järjestelmään. Luvun lopussa esitetään yhteenveto ongelmista ja arvioidaan niiden vaikutuksia ja riskejä.

5.1 Aineiston toimitus

Arkistointiprojekti alkaa aineiston toimituksesta ja sen analysoinnista. Projekteissa A ja B aineisto toimitettiin poimimattomana. Projekteissa C ja D aineisto toimitettiin valmiiksi poimittuna.

Potilastietojärjestelmän arkkitehtuuri voi olla sellainen, että asiakkaan ei ole mahdollista päästä käsiksi tietokantaan itse, vaan toimenpiteet joudutaan tilaamaan järjestelmätoimittajalta. Ennen ohjelmistot saattoivat käyttää yhtä tietokantapalvelinta yhtä toimipistettä kohden, mutta nykyään ohjelmistot ovat enemmän keskitettyjä. Keskitetyissä ratkaisuissa sama palvelin voi palvella monia toimipisteitä, jolloin siinä voi olla myös useamman toimipisteen potilastietoja. Vaihtoehdoksi on tullut myös ohjelmistojen ostaminen palveluna, jolloin tiedot sijaitsevat järjestelmätoimittajan palvelimilla.

On mahdollista, että järjestelmätoimittaja ei ole yhteistyöhaluinen aineiston poiminnan avustamisessa, sillä silloin järjestelmätoimittaja mahdollistaa järjestelmästä luopumisen ja voi menettää asiakkaan. On esimerkiksi mahdollista, että toimittaja on asentanut järjestelmän siten, että käyttäjät eivät pääse käsiksi varsinaiseen järjestelmän tietoon, kuten tietokantaan. Yhteistyöhaluttomuus voi johtaa viivästymiseen ja aiheuttaa lisää ongelmia. Jos järjestelmätoimittaja avustaa poiminnassa, se voi vaikuttaa aikatauluihin ja aiheuttaa riippuvuuden ulkoisesta tekijästä, vaikka kaikki sujuisi ongelmitta.

Vaikka järjestelmätoimittaja avustaisi aineiston toimituksessa, lähdejärjestelmä voi perustua sellaiseen ratkaisuun, että poimitun aineiston käyttöönottoon liittyy ongelmia. Järjestelmä voi olla rakennettu suljetun ohjelmistoratkaisun varaan, jonka analysointi ei onnistu ulkopuoliselta. Mikäli käytössä on tietokantajärjestelmä, jonka käyttämiseksi ei löydy sopivia ohjelmistokomponentteja, aineiston poiminnassa joudutaan miettimään

muita vaihtoehtoja. Myös tietokantajärjestelmän hinnoittelu voi olla korkea, mikä täytyy huomioida arkistointia aloitettaessa.

Arkistoitavat potilastietojärjestelmät ovat olleet käytössä yleensä hyvin kauan, joten niissä käytettävät järjestelmät ovat yleensä jo vanhentuneita. Vanhentuminen aiheuttaa ongelmia esimerkiksi tietokantajärjestelmien kanssa toimimiseen. Vaikka järjestelmän tietokannasta saataisiin tehtyä vedos, sen lataaminen nykyaikaiseen tietokantaohjelmistoon ei onnistu. Tämä voi johtua esimerkiksi siitä, että järjestelmät eivät ole taaksepäin yhteensopivia eikä alkuperäistä ohjelmistoversiota ole enää mahdollista löytää. Lisäksi vedoksen muodostaminen järjestelmästä siten, että sen käyttäminen onnistuu muissakin ympäristöissä, voi olla vaikeaa. Esimerkiksi B-projektin tietokantaohjelmisto oli niin vanha, että sen ajurin löytäminen oli todella haastavaa.

Järjestelmien ikä vaikeuttaa myös oikeiden asiantuntijoiden löytämistä. Esimerkiksi B-projektissa järjestelmä oli poistettu käytöstä niin kauan sitten, että asiantuntijoiden löytäminen ei ollut helppoa. Asiantuntijoita tarvittiin esimerkiksi selvittämään, kuinka järjestelmän eri ominaisuuksia oli käytetty ja minkälaisia virhetilanteita niissä oli havaittu.

Erot käyttöjärjestelmien tai ohjelmistoympäristöjen välillä voivat aiheuttaa haasteita. Tietokantaohjelmisto voi esimerkiksi toimia sekä Unix-pohjaisessa Linuxissa ja Windows järjestelmissä, mutta vedokset näiden käyttöjärjestelmien välillä eivät välttämättä ole yhteensopivia. Vedoksen tuonnin jälkeen ongelmana on myös sen varmistaminen, että vedos vastaa alkuperäistä tietokantaa eikä tietoa ole hävinnyt. Näitä ongelmia ei havaittu käsiteltävissä projekteissa, mutta tietokannan palauttaminen vedoksesta on joissain arkistointiprojekteissa ollut työlästä eri ympäristöistä johtuvista syistä.

Tietokantavedoksen saattaminen alkuperäistä tietokantaa vastaavaan muotoon saattaa vaatia monia työvaiheita eikä ole järkevää, että jokainen kehittäjä joutuu toistamaan samat vaiheet. Tämä ongelma voidaan ratkaista esimerkiksi asentamalla tietokanta virtuaaliympäristöön, jota kaikki käyttäjät voivat käyttää tai asentaa tietokanta keskitetysti palvelimelle, jolloin kehittäjät voivat muodostaa yhteyden samaan tietokantaan.

Projekteissa C ja D aineisto saatiin valmiiksi poimittuna. Projektin C perusteella oletettiin, että myös projektissa D aineisto toimitettaisiin samassa muodossa. Aineiston poiminnassa oli kuitenkin eroja, jotka viivästyttivät projektia huomattavasti.

5.2 Tietokannan analysointi

Projekteissa havaittiin, että eri järjestelmien välillä voi olla hyvin suuria eroja ja niiden toiminnan ymmärtäminen voi olla hyvin haastavaa. Jotta tietokannasta saadaan poimitua oleelliset osat, on sen sisältö ja rakenne ymmärrettävä riittävän hyvin.

Tietokannan sisältö ja rakenne voidaan selvittää takaisinmallintamalla [38]. Siinä pyritään esimerkiksi selvittämään tietokannan taulujen riippuvuudet, jotta niiden suhteista voidaan päätellä taulujen kuvaamien olioiden koostumuksia. Esimerkiksi potilaan terveystietomus voi koostua useista tietokantariveistä, jotka viittaavat yhteen kertomukseen. Kun nämä suhteet ymmärretään, saadaan tietokannan sisällöstä muodostettua potilaan kaikki potilaskertomukset.

Käänteismallinnusta voidaan tehdä esimerkiksi tutkimalla vierasavaimilla toteutettuja taulujen välisiä viittauksia [39]. kaikki tietokantajärjestelmät eivät kuitenkaan tue vierasavainten käyttöä. Esimerkiksi projektin B tietokanta ei käyttänyt viittauksia, vaan analysoinnissa jouduttiin turvautumaan sarakkeiden nimien avulla tapahtuvaan analysointiin, jossa oletettiin, että eri tauluissa käytetyt saman nimiset sarakkeet tarkoittivat samaa asiaa. Aina viittauksia ei voitu päätellä nimienkään avulla, vaan viittaukset jouduttiin paikantamaan oletettujen avainsarakkeiden sisältöjä yhdistelemällä.

Lisäksi apuna käytettiin työkalua, jolla nähtiin järjestelmän käyttämät kyselyt eri toimintoja suorittaessa. Työkalun käyttö auttoi merkittävästi järjestelmän analysoinnissa, sillä tietokantakyselyiden perusteella voitiin tutkia, miten järjestelmä käytti tietokannan eri tauluja. Kyselyiden näkeminen auttaa erityisesti, kun tietokantojen taulujen nimistä ei voida päätellä niiden tarkoitusta. Silloin järjestelmällä voidaan tehdä asioita, joiden oletetaan liittyvän tietokantataulun sisältöön ja kyselyistä nähdään, oliko taulun oletettu tarkoitus arvioitu oikein.

Vierasavaimet on voitu toteuttaa myös ohjelman sisällä, ilman tietokantajärjestelmän tukea [39]. Tämä tarkoittaa, että tietokannan rivin arvosta muodostetaan ohjelmiston sisäisten sääntöjen avulla toinen arvo, jolla viitattava rivi ja/tai taulu löydetään. Ohjelmiston sisäiset operaatiot ovatkin yksi suurimmista ongelmista käänteismallinnuksen taustalla. Projektissa B vierasavaimen kohdetaulu määräytyi kyseisen vierasavaimen jakojäännöksen perusteella siten, että jakojäännös kertoi viitattavan kohteen tyyppin. Toisessa tapauksessa negatiiviset vierasavaimet osoittivat eri tauluun kuin positiiviset. Negatiivisen avaimen itseisarvo määräsi käytettävän tietokantarivin. Järjestelmän käyttämien kyselyiden tutkiminen ei auta piilotettujen vierasavaimien etsinnässä, sillä viittauskohteen muodostuslogiikka on järjestelmän sisällä eikä kyselyssä. Projektin B jakojäännöksen tarkoitus selvisi käyttöohjeen perusteella, joten dokumentaatiosta voi olla paljon hyötyä.

Vierasavainten puuttuminen tarkoittaa myös, että tietokanta ei voi tarkistaa viitteiden eheyttä, jolloin sinne voi syntyä orporivejä. Orporivien ongelmana on, että ne voivat sisältää tietoa, mutta siihen ei viitata missään, joten tieto ei silloin näy järjestelmälle. Täl-

laista sisältöä ei voida myöskään havaita ohjelmiston käyttöliittymästä. B-projektissa löydettiin tilanne, jossa järjestelmä oletti, että tietokannan rivi viittaa yhteen riviin toisessa taulussa, mutta kohdetaulusta löytyikin useampi vastaava rivi, jolloin järjestelmä näytti vain ensimmäisen rivin tiedot. Tällainen tilanne voi syntyä esimerkiksi ohjelmistovirheen seurauksena, mutta virhe ei näy järjestelmän ulkopuolelle esimerkiksi käyttöliittymään.

Projektin B järjestelmässä oli suoritettu aikaisemmin versiopäivitys, jossa vanhan tietokantataulun sisältö oli osittain kopioitu uuteen tauluun. Vanhan tietokantataulun sisältöä ei voitu tarkastella käyttöliittymällä, mutta sen kaikkea tietoa ei kuitenkaan löytynyt uudesta taulusta. Jos tavallinen loppukäyttäjä ei voi nähdä tietoa, joudutaan miettimään, että onko tieto silloin arkistoitava. Tällaiset tapaukset joudutaan arvioimaan aina yksittelen asiakkaan kanssa. Toisaalta tieto on saatettu kirjata, mutta siirto uuteen tietokantatauluun ei onnistunut tai tieto on jätetty siirtämättä tarkoituksella.

Tietokantasuunnittelussa oleellista on normalisointi. Normalisoinnin tavoitteena on esimerkiksi varmistaa, että sama tieto on vain yhdessä paikassa. Kun tätä oletusta rikotaan ja sama tieto löytyy useammasta paikasta, se voi olla keskenään ristiriitaista. Normalisoinnin puute aiheutti ongelmia projektissa B. Projektin B tietokannasta löytyi kaksi taulua, jotka sisälsivät potilaiden syntymäpäiviä. Tauluista löydettiin tapauksia, joissa henkilön syntymäpäivät erosivat taulujen välillä. Arkistointiprojektin kannalta tämä tarkoittaa sitä, että on kehitettävä sääntö, kumpaa tietoa käytetään. Sääntö voi perustua esimerkiksi ohjelmiston toiminnan havainnointiin tai se voidaan muodostaa siten, että huomioidaan viimeisin tieto. Mikäli tietojen välillä havaitaan ristiriita, eikä sääntöä voida muodostaa, voi olla syytä käyttää molempia tietoja. Tämänkaltaisia tilanteita voivat olla esimerkiksi henkilön terveystietojen päivittymien useaan paikkaan.

Tietoa voi olla useassa paikassa, vaikka kyseessä ei ole normalisointiongelma. Versiopäivitysten myötä tietokannassa voi olla vanhoja tauluja, joita ohjelmisto ei enää käytä. Vaikka analysoinnin perusteella voi vaikuttaa siltä, että ohjelmisto ei käytä näitä tauluja, sen todistaminen saattaa olla haastavaa. Erityisesti silloin, kun ohjelmistoa ei ole käytettävissä, voi olla mahdotonta osoittaa, mitä tietoa ohjelmisto käyttää ja mitä ei. Projektissa B löytyi tauluja, joissa oli sisältöä, mutta joita järjestelmä ei näyttänyt käyttävän.

Aikaisempien vastaavien järjestelmien analyysien perusteella ei välttämättä voida vetää suoraan johtopäätöksiä uuden järjestelmän tietokannan sisällöstä, sillä versiomuutosten takia uuden järjestelmän rakenteet ovat saattaneet vaihtua kokonaan. Muutoksia voivat aiheuttaa myös paikalliset korjaukset ja ominaisuudet, joita tämänkaltaisiin ohjelmistoihin kehitetään. Versiomuutokset ovat saattaneet aiheuttaa sellaisia muutoksia, että jotkin

vanhoista tiedoista eivät enää toimi oikein, mutta niitä ei ole käytetty vuosiin, joten virhettä ei ole havaittu. Projektissa B käytettävä lomakeominaisuus mahdollisti lomakkeen rakenteen muokkaamisen siten, että muutos vaikutti myös aikaisempiin lomakkeisiin. Jos lomakkeesta esimerkiksi poistettiin jokin tietokenttä, sitä ei enää nähnyt myöskään vanhoista lomakkeista. Järjestelmässä ei ollut käytössä versiointia, jolla ongelma oltiin ratkaistu.

5.3 Järjestelmän analysointi

Järjestelmää voidaan analysoida takaisinmallintamalla. Takaisinmallinnuksessa pyritään ymmärtää järjestelmän toiminnallisuus tutkimalla sen käyttäytymistä eri syötteillä, analysoida koodia tai tutkimalla sen kuvauksia ja dokumentteja. Analysoitavana koodina voidaan käyttää lähdekoodia, esikäännettyä koodia kuten Javan tavukoodia tai konekielellistä koodia. [38]

Tässä työssä analysoidut projektit tehtiin järjestelmään nähden täysin ulkopuolisina, eli arkistojalla ei ollut mitään tietoa järjestelmän sisäisestä toiminnasta eikä käytettävissä ollut tukea järjestelmän toimittajalta. Analysoinnit tehtiin ilman järjestelmän lähdekoodia, eikä jatkossakaan ole oletettavaa, että lähdekoodeja saadaan käytettäväksi. B-projektissa analysointiin käytettiin järjestelmästä löytyneitä käyttöohjeita ja järjestelmän asennuksessa syntyneitä dokumentteja. Käyttöohjeet olivat huomattavasti teknisempiä, kuin mitä loppukäyttäjän ohjeilta olettaisi, joten niistä oli suuri hyöty järjestelmän analysoinnissa.

Järjestelmän analysointia voidaan tehdä esimerkiksi käyttämällä järjestelmää ja havainnoimalla järjestelmän ulostuloja käyttöliittymästä ja mahdollisesti tietokannasta. Jos lisäksi nähdään tietokantakyselyt, joita järjestelmä käyttää, järjestelmän toiminta voi olla mahdollista ymmärtää hyvin. Toiminnan ja havaintojen perusteella voidaan muodostaa sääntöjä, joiden perusteella järjestelmän oletetaan toimivan. Näiden sääntöjen löytäminen on tärkeää, mikäli tietokannan käänteismallinnuksessa löydetään tietoja, joiden merkitystä tai sisältöä ei ymmärretä. Jos käänteismallinnus tehtäisiin täydellisesti, näiden sääntöjen avulla voitaisiin muodostaa täydellinen kopio alkuperäisestä järjestelmästä.

Käytännössä säännöt eivät kuvaa aina täydellisesti tutkittua järjestelmää. Järjestelmien suunnittelussa on aina huomioitava rajatapaukset, joita sen käytössä voi tapahtua. Analysointivaiheessa yleisten sääntöjen lisäksi pitäisi pystyä myös tunnistamaan rajatapauksia, joissa järjestelmä ei toimi havaitun säännön perusteella. Voi olla esimerkiksi tilanne, jossa järjestelmä käsittelee vanhaa lomakemuotoa eri tavoin kuin uutta, mutta vanhoja

lomakemuotoja löytyy vain vähän. Silloin on mahdollista, että lomakkeen käsittelyyn liittyvää rajatapausta ei löydy. Tästä voi seurata, että vanhan lomakemuodon käsittelyssä tapahtuu virhe, joka aiheuttaa sen sisällön katoamisen tai virheellisyyden, mutta tilannetta ei huomata, sillä virhettä ei löydetty analysointivaiheessa. Automaattisella aineiston läpikäynnillä tällainen virhe voitaisiin havaita.

Järjestelmän analysointia voidaan tehdä testiaineistolla tai oikealla aineistolla. Testiaineiston ongelmana on, että se ei välttämättä vastaa oikeassa käytössä syntynyttä aineistoa, jolloin sen perusteella ei voida selvittää järjestelmän toimintaa. Testiaineisto voi olla esimerkiksi järjestelmätoimittajan tuottamaa, jolloin se eroaa hyvin paljon käytön aikana syntyneestä aineistosta. Testiaineisto voi myös olla liian suppea, jolloin sen analysoinnilla ei saada tutkittua kaikkia tietotyyppejä. Toisena vaihtoehtona on käyttää oikeaa järjestelmästä löytyvää aineistoa. Potilastietoa pitäisi katsoa vain, kun siihen on tarve, mutta analysointia ei välttämättä pysty suorittamaan ilman oikeaa aineistoa. Silloin voi olla tarpeellista poistaa aineistosta henkilötiedot ja viitteet oikeisiin henkilöihin pseudonymisoimalla se, kuten aliluvussa 3.2.2 kuvattiin.

Käyttöliittymä voi muodostaa tietokannasta hyvin erilaisen esitysmuodon, jonka logiikkaa ei voida päätellä ilman järjestelmän käyttöä. Pelkällä tietokannalla ei tässä tapauksessa tee mitään. Esimerkiksi projektissa B järjestelmä muodosti näytettävän lomakkeen dynaamisesti tietokannassa säilötyn lomakkeen rakenteen ja sisällön perusteella. Lomakkeen rakenne riippui lomakkeen tyypistä, mutta lomakkeen sisältö vaihteli lomakeinstanssin mukaan. Järjestelmän mukana saatiin lomaketyökalu, jolla uusia lomakkeita voitiin muodostaa. Kyseisen työkalun avulla käänteismallinnettiin lomakkeen säilytysmuoto, jotta vastaavia lomakkeita saatiin tuotettua arkistoitavaksi ilman järjestelmän käyttöä.

Potilastiedon arkistoon muodostetaan XML-asiakirjoja, joiden esitysmuoto on varsin erilainen potilastietojärjestelmiin verrattuna. Asiakirjamuoto rajoittaa tietojen esitysmuotoa arkistoinnin jälkeen. Jotkut toiminnallisuudet eivät myöskään toimi asiakirjamuodossa. Esimerkki tällaisesta ovat vieritettävät kentät tai interaktiiviset kaaviot, joiden sisältöä käyttäjä voi tarkastella eri tavoin. Esimerkiksi projektissa B lomakkeiden kenttiä voitiin vierittää eli ne olivat interaktiivisia. Kun tällaiset lomakkeet viedään asiakirjamuotoon, niihin voi jäädä esimerkiksi kenttiä, joiden sisältö ei näy kokonaan, sillä niitä ei voi vierittää.

Toinen esimerkki on graafit, joissa voi olla selitteitä, jotka tulevat näkyviin hiirellä kohdistettaessa. Vaihtoehtoina on joko jättää selitteet näyttämättä tai näyttää kaikki selitteet samaan aikaan. Lomakkeesta voi tulla hyvin sekava, jos kaikki selitteet ovat näkyvissä

samaan aikaan, mutta niitä ei voi myöskään jättää pois. Esitystapaa voidaan myös joutua muuttamaan, jos arkistoidaan esimerkiksi hammaskarttoja, joissa jokaiselle hampaalle aukeaa oma selite.

5.4 Tiedon luokittelu ja rikastus

Kun lähtötietojen sisältö on analysoitu, on selvittettävä, mikä tieto on arkistoitavaa ja mikä tieto on säilytettävä muualla. Potilastietoa ei voida määritellä kovin tarkasti. Tarkan määrityksen puuttuminen aiheuttaa ongelmia, kun mietitään mitä arkistoidaan ja mitä ei. Potilastietojärjestelmiin voidaan kirjata asioita, jotka muistuttavat potilastietoa, mutta eivät sitä välttämättä ole. Projektissa B aineistosta löytyi tiedot potilaiden osastolle kirjaamisesta ja uloskirjaamisesta. Tiedot näyttivät potilastiedoilta, mutta asiakas määritteli ne tilastollisiksi tiedoiksi, joita ei arkistoida. Projektissa C jätettiin arkistoimatta tiedot, jotka olivat hallinnollisia lomakkeita.

Potilastiedon lisäksi järjestelmät sisältävät tietoja, jotka on säilytettävä esimerkiksi lain [3] tai asiakkaan vaatimuksesta. Lain vaatimuksesta säilytettäviä tietoja ovat esimerkiksi käyttölokot, jotka kertovat kuka on käyttänyt järjestelmää ja mitä kyseinen käyttäjä on katsonut. Asiakkaan vaatimuksesta arkistoitavia asioita voivat olla esimerkiksi laskutukseen tai työntekijöiden työajanseurantaan liittyvät asiat. Vaikka näitä asioita ei arkistoida Kantaan, nekin on huomioitava arkistointiprojektissa.

Potilastiedon arkistoinnissa tietoa rikastetaan lisäämällä siihen luvussa 3.1.2 kuvatut näkymät sekä tarpeen mukaan vaiheet ja otsikot. Näiden tietojen pakollisuus riippuu sisällön muodosta. Vanhoissa tiedoissa sisältönä voidaan käyttää näyttömuotoa tai vanhojen potilastietojen erikoistatapausta. Näyttömuodossa sisällölle määritetään sopiva hoitoprosessin vaihe sekä otsikko. Näitä käsitteitä ei kuitenkaan ole välttämättä ollut olemassa, kun potilastietojärjestelmää kehitettiin, joten niiden tuottaminen on suoritettava saatavissa olevien tietojen perusteella.

Näkymän tarkoituksena on luokitella kirjauksen sisältöä ja auttaa ryhmittelemään samaan aihepiiriin liittyviä kirjauksia helpommin löydettäväksi. Vaikka näkymä-käsitettä ei olisi ollut olemassa, kun järjestelmää käytettiin, voidaan näkymä yleensä päätellä, mikäli tiedot on ryhmitelty loogisesti. Järjestelmässä voi esimerkiksi olla erilaisia ominaisuuksia, joilla tuotetaan tietynlaisia kirjauksia. Esimerkkejä näistä ominaisuuksista ovat päivittäismerkinnät vuodeosastolta sekä terveyskeskuskäynnin kirjaus. Kontekstista voidaan siis päätellä, mihin kategoriaan tieto kuuluu. On kuitenkin mahdollista, että järjestelmä sijoittaa kaikki tiedot samaan kategoriaan, jolloin erottelua ei voida tehdä. Lisäksi on mahdollista, että tieto on alun perin kirjattu väärään paikkaan, jolloin tieto luokitellaan väärin.

Projektissa B osa näkymistä pääteltiin suoraan arkistoitavan tiedon perusteella. Esimerkiksi työterveysmerkinnät menivät TYÖ-näkymälle ja seurantamerkinnät HOI-näkymälle. Osa näkymistä pääteltiin tekstin sisällön perusteella etsien sieltä avainsanoja, jotka kuvasivat sisältöä. Avainsanoja saattoivat olla esimerkiksi rokotus tai osteoporoosi. Projektin B järjestelmän lomakkeet luokiteltiin kukin omaan näkymäänsä ja myös asiakas sai vaikuttaa luokitteluun.

Projekteissa A, C ja D näkymät perustuivat tekstin avainsanoihin. Avainsanoja muodostettiin sisällön perusteella ja asiakkaan palautteen avulla. Näissä järjestelmissä sisällöllä ei ollut sellaisia metatietoja, jotka olisivat auttaneet luokittelussa, vaan kaikki tieto näytti samalta.

Ominaisuuksien avulla voidaan päätellä myös kirjauksen vaihe tai otsikko. Käytössä on esimerkiksi saattanut olla paikallisia otsikoita, joita lääkärit ovat käyttäneet kirjauksia muodostaessaan ja näitä otsikoita voidaan yhdistää kansallisiin otsikoihin. Otsikoita voidaan koittaa myös päätellä kirjauksen sisällöstä avainsanahaun avulla. Hoitoprosessin vaihettakin voidaan yrittää muodostaa tietojen perusteella, mikäli potilastietojärjestelmä on mahdollistanut tietojen kirjaamisen vaiheittain hoitoprosessin edetessä.

Myös hoitoprosessin vaiheet ja otsikot pääteltiin tekstin sisällön perusteella. Projektissa B kirjat sisälsivät otsikoita, mutta ne olivat paikallisesti määriteltyjä, joten ne kohdistettiin kansallisiin otsikoihin, jolloin niitä voitiin käyttää suoraan. Otsikoille määriteltiin lisäksi vastaavuus hoitoprosessin vaiheeseen, jolloin samoja vastaavuuksia voitiin käyttää seuraavissa projekteissa.

Mikäli edellä mainittuja tietoja ei voida muodostaa, voi olla hyödyllisempää käyttää vanhojen potilastietojen erikoistapausta. Projektissa B käytettiin rakenteetonta muotoa, kun kirjaukselle ei voitu muodostaa otsikoita. Rakenteetonta muotoa käytettiin myös, kun projekteissa muodostettiin pdf-muotoisia asiakirjoja.

5.5 Transformaatio

Transformaatiovaiheessa ongelmat ovat yhteisiä kaikille projektityypeille. Aineistopaiminta on nyt suoritettu ja aineisto on analysoitu ja jäljellä on varsinainen arkistoitavan aineiston transformaatio. Transformaatioissa aineistosta muodostetaan analyysiin perustuen asiakirjoja, jotka vastaavat parhaiten järjestelmän sisältöä.

Projekteissa käytettiin järjestelmää, joka muodosti CDA asiakirjat ja yksinkertaisti asiakirjojen tuottamista esimerkiksi hakemalla niihin automaattisesti oikeat koodit ja täyttämällä monimutkaiset otsikkotiedot, joten asiakirjojen muodostaminen ei aiheuttanut suu-

ria ongelmia projektiin. Mikäli valmista asiakirjojen muodostusjärjestelmää ei olisi käytettävissä, asiakirjojen muodostus veisi huomattavasti enemmän aikaa, sillä kuten edellisistä luvuista nähdään, asiakirjan rakenne on melko haastava ja monimutkainen. Toisaalta asiakirjan rakenne on hyvin määritelty, joten sen toimintalogiikka ei vaihtelee eri järjestelmien välillä.

Potilastiedon arkistoon viedään yleensä tekstikirjauksia, mutta vanhojen tietojen kohdalla sallitaan myös rakenteettomien tietojen arkistointi. Se on ainoa mahdollisuus, kun arkistoitava merkintä on PDF-muotoinen. Silloin PDF-asiakirja sijoitetaan CDA-asiakirjaan PDF-A-1b muodossa. Muoto on tarkoitettu asiakirjojen pysyvään arkistointiin siten, että ne ovat aina saatavissa eikä niiden esitysmuoto vaihtelee ohjelmien välillä [40].

Vaikka kyseinen PDF-muoto on määritelty tarkasti, eri toimijat tulkitsevat samoja määrittäyksiä eri tavoin. Projekteissa C ja D havaittiin, että välillä käytetty validointityökalu ei hyväksy tuotettua tiedostoa vaan siihen on jouduttu tekemään korjauksia, vaikka tiedoston versio olikin PDF-A-1b. Korjaukset liittyvät yleensä ominaisuuksiin, jotka PDF-A-1b määrittäminen sallii, mutta ei vaadi.

5.6 Tuotantoonvienti

Huolimatta siitä kuinka hyvin projektin analyysi ja muut vaiheet toteutetaan, saattaa tuotetussa aineistossa olla silti puutteita. Esimerkiksi tietokanta-analyysissä voidaan olettaa, että jokin viittaus on yksi-yhteen, mutta todellisuudessa viittaus kohdistuu useampaan kohteeseen, jonka seurauksena tuotetusta aineistosta puuttuu tietoja. Pahimmassa tapauksessa löydetään systemaattinen virhe, jonka takia koko aineisto on virheellinen. Havaitut virheet tulisi korjata tuotettuun aineistoon, samoin kuin lääkärit voivat korjata virheelliset kirjaukset ne havaittuaan. Projektin tuotantoonviennin jälkeen havaittujen virheiden korjaaminen ei kuitenkaan ole helppoa.

Tavallisesti lääkäri voisi korjata virheellisen merkinnän etsimällä sen järjestelmästä ja korjaamalla sen oikeaksi. Vanhojen tietojen arkistoinnissa asiakirjat arkistoidaan Kelan järjestelmään, joka ei salli tämänkaltaista muokkaamista. Sen sijaan asiakirjasta arkistoidaan uusi versio ja vanha asiakirja mitätöidään. Se tarkoittaa, että korjaukseen tarvitaan alkuperäinen mitätöitävä asiakirja ja lisäksi mahdollisesti aineisto, josta asiakirja muodostettiin. Korjaus tehdään ottamalla vanhan asiakirjan sisältö pohjaksi ja muokkaamalla siihen lisäykset ja poistot.

Systemaattisen virheen kohdalla jokainen virheellinen asiakirja joudutaan etsimään ja korjaamaan yksitellen, joten pitäisi olla mahdollista selvittää mihin asiakirjoihin havaittu virhe vaikuttaa. Jotta lähtöaineisto voidaan yhdistää tuotettuun aineistoon, on pidettävä

kirjaa siitä, mistä asiakirjan sisältö on koostettu. Luotu asiakirja on siis kyettävä yhdistämään lähtöaineistoon. Luotujen asiakirjojen tunnisteet on siis pidettävä tallessa, jotta niiden avulla voidaan myöhemmin suorittaa massakorjauksia.

Korjauksia varten on säilyttävä myös lähtöaineisto. Projektin tavoitteena on kuitenkin ylimääräisistä järjestelmistä luopuminen, joten virheiden korjaaminen on ristiriidassa projektin tavoitteiden kanssa. Lähtöjärjestelmää ei kuitenkaan tarvitse säilyttää kokonaan, vaan riittää, että siitä säilytetään arkistoinnissa käytetty aineisto. Säilyttäminen jää tilaajan vastuulle, sillä muilla ei ole oikeutta säilyttää aineistoa pitkäkestoisesti. Täytyy myös selvittää, kuinka kauan aineistoa voidaan säilyttää.

Migraation yhtenä vaiheena on yleensä varmistaa, että migraatio onnistui eli kaikki siirrettiin onnistuneesti. Myös arkistointiprojektissa tämä on tärkeä vaihe, sillä arkistoinnin jälkeen alkuperäinen aineisto saatetaan poistaa, jolloin siirtymätön tieto katoaa pysyvästi. Migraatioprojekteissa siirretään tietoja tietokannasta toiseen, jolloin migraation onnistuminen voidaan varmistaa vertaamalla lähde- ja kohdetietokannan rivejä, joiden sisältöjen pitäisi vastata toisiaan. Jos migraatiossa on esimerkiksi siirretty lähdejärjestelmästä asiakastietoja, voidaan tietokantakyselyillä varmistaa, että lähdejärjestelmästä löytyneet asiakkaat löytyvät myös kohdejärjestelmästä vastaavilla tiedoilla. Tällaista vertailua voidaan suorittaa esimerkiksi satunnaisesti valituille asiakkaille.

Sen sijaan arkistointiprojektissa siirrettävät tiedot koostetaan asiakirjoiksi, jolloin lähdejärjestelmässä esiintyneet tiedot, kuten tietokantarivit, muunnetaan kohdejärjestelmän asiakirjoiksi, jolloin ei ole yksinkertaista keinoa suorittaa kyselyitä vastaavuuksien vertailumiseksi. Ongelma on, että kohdejärjestelmän tietojen alkuperä ja suhde lähtötietoihin hämärtyy, joten automaattisen vertailun suorittaminen on haastavaa.

Arkistoinnin kattavuutta voidaan tutkia manuaalisesti vertaamalla lähdejärjestelmän tietoja tuotettuun aineistoon. Kaiken tiedon vertailu kestää kuitenkin hyvin kauan, joten vertailtavaksi valitaan yleensä vain pieni osajoukko, joten on riskinä, että jokin harvemmin esiintyvä virhe jää löytämättä.

5.7 Testaus

Testauksessa pyritään varmistamaan, että ohjelma toimii sille asetettujen tavoitteiden mukaisesti. Datamigraatioprosessin testauksessa varmistetaan, että aineiston poiminta ja transformaatio onnistuivat. Nykyisessä prosessissa kyseisten vaiheiden testaaminen ei onnistu toisistaan erillään, vaan testaukseen käytetään poimittua ja transformoitua aineistoa.

Aineiston poiminnan onnistumisen varmistamiseen joudutaan käyttämään transformoitua aineistoa, mikä tekee siitä hankalampaa kuin pelkästään poimitun aineiston testaamisesta. Tämä johtuu siitä, että transformoidulla aineistolla ei ole enää alkuperäistä aineistoa vastaavaa rakennetta ja siitä on jo saatettu suodattaa tietoja pois, minkä takia sen vertaaminen alkuperäiseen aineistoon ei aina onnistu lainkaan tai voi olla hyvin haastavaa.

Transformaatiota testataan manuaalisesti vertailemalla potilastietojärjestelmän tietoja tuotettuihin tietoihin. Automaattitestaus on haastavaa, sillä projektin yhtenä tavoitteena on muodostaa säännöt, joilla potilastietojärjestelmä on toiminut ja testauksen onnistuminen riippuu näiden sääntöjen oikeellisuudesta. Tällä tarkoitetaan sitä, että ei ole mahdollista tarkistaa noudattaako tutkittava järjestelmä muodostettua sääntöä, ilman että kaikki tapaukset tarkastetaan käsin. Useat virheet havaitaan manuaalisella testauksella.

Manuaalisen testauksen vuoksi koko aineistoa ei ole mahdollista käydä läpi, vaan testaukseen poimitaan pieni osajoukko aineistosta. Testauksessa on myös käytettävä alkuperäistä aineistoa sillä testausta varten kehitetty aineisto ei mahdollista alkuperäistä aineistoa vastaavaa testausta. Alkuperäisen aineiston käyttäminen vaikuttaa tietoturva-vaatimuksiin jo prosessin kehitysvaiheessa.

Automaattisten nopeiden testien yhtenä etuna olisi regressiotestauksen mahdollistuminen, eli vanhojen virheiden havaitseminen. Manuaalitestauksessa regressiotestaus joudutaan huomioimaan siten, että testauksen edetessä muodostuu kasvava joukko testitapauksia, jotka joudutaan käymään aina uudelleen läpi. Testauksen työmäärän kasvassa kasvaa myös riski siihen, että testauksessa ei havaita virhettä, sillä testaaja ei ole riittävän huolellinen tai testaukseen ei riitä tarpeeksi aikaa.

5.8 Muut ongelmat

Edellä kerrottujen haasteiden lisäksi on olemassa pienempiä haasteita, jotka liittyvät tietojen erotteluun. Arkistoitavat tiedot kuuluvat aina yhdelle organisaatiolle. Arkistointiprojektiin voidaan lähteä esimerkiksi silloin, kuin kuntayhtymän hajotessa sen potilastiedot arkistoidaan. Potilastiedot on silloin jaettava eri rekistereihin järjestelmästä löytyvien tietojen perusteella. Jako voidaan tehdä esimerkiksi potilaiden asuinkuntien mukaan [18].

Potilastietojärjestelmästä voi löytyä julkisen, yksityisen sekä työterveyshuollon potilastietoja. Ne kaikki kuuluvat arkistoinnissa omiin rekistereihinsä. Työterveystiedot pitäisi lisäksi voida jakaa alirekistereihin työnantajan mukaan. Esimerkiksi työterveyden ja julkisen puolen terveystietojen erottaminen voi olla haastavaa, jos sama henkilö on käynyt samassa hoitopaikassa sekä julkisella että työterveyden vastaanotolla. Silloin joudutaan

miettimään miten esimerkiksi hoitoketjun katkeaminen aineiston jaon seurauksena arkistoinnissa vaikuttaa hoitotilanteisiin myöhemmin.

5.9 Yhteenveto ongelmista

Taulukossa 1 on esitetty yhteenveto arkistoinnin haasteista ja niiden vaikutuksista projektin läpivientiin aineistotoimituksen mukaan. Taulukosta nähdään, että eniten ongelmia kohdataan, kun arkistoiija joutuu poimimaan aineiston itse. Tällöin tiedon analysointiin kuluu paljon aikaa ja on myös riski, että analysoinnissa tehdään oletuksia, jotka eivät pidäkään paikkaansa, jolloin tietoja saattaa jäädä arkistoimatta.

Taulukko 1. Yhteenveto arkistoinnin haasteista ja vaikutuksista aineistotoimituksen tyypin mukaan.

Prosessin vaihe	Haasteet (poimittu aineisto) suluissa vaikutus projektiin	Haasteet (poimimaton aineisto) suluissa vaikutus projektiin
Aineiston toimitus	<ul style="list-style-type: none"> - Riippuvuus osapuolesta, joka poimii aineiston (keston piteneminen, tietoja jää poimimatta) - Järjestelmä on vanha (Asiantuntijoiden löytäminen on hankalaa) 	<ul style="list-style-type: none"> - Järjestelmätoimittajan yhteistyöhaluttomuus (keston piteneminen, projektia ei voi tehdä) - Järjestelmä on vanha (yhteensopivuusongelma esim. tietokantojen pystytyksessä, asiantuntijoiden löytäminen hankalaa) - Erot ohjelmistoympäristöjen välillä, kuten käyttöjärjestelmät (keston piteneminen, projektia ei voi tehdä) - Ohjelmistolisenssit (kustannukset)
Tietokannan analysointi	-	<ul style="list-style-type: none"> - Lähdetietokanta on huonosti suunniteltu (keston piteneminen, tietoja jää poimimatta, työvoimaresurssien tarve kasvaa) - Lähdetietokantaan on tehty muutoksia (keston piteneminen, tietoja jää poimimatta, työvoimaresurssien tarve kasvaa)
Järjestelmän analysointi	-	<ul style="list-style-type: none"> - Lähdejärjestelmän monimutkaisuus (keston piteneminen, tietoja jää poimimatta, työvoimaresurssien tarve kasvaa, poiminta joudutaan ulkoistamaan) - Käyttäjävirheet (tietoja jää poimimatta)
Tiedon luokittelu ja rikastus	<ul style="list-style-type: none"> - Potilastiedon tunnistaminen (keston piteneminen, tietoja jää poimimatta) - Metatietojen muodostaminen (keston piteneminen, työvoimaresurssien tarve kasvaa) 	
Muunnosvaihe	<ul style="list-style-type: none"> - PDF-muodon standardi (keston piteneminen) 	<ul style="list-style-type: none"> - PDF-muodon standardi (keston piteneminen) - Kaikki tieto ei sovi asiakirjamuotoon (Tiedon käytettävyys huononee, tietoja jää poimimatta)
Tuotantoonviennin ongelma	<ul style="list-style-type: none"> - Virheiden korjaaminen hankalaa (työvoimaresurssien tarve kasvaa, keston piteneminen) - Arkistoinnin kattavuuden varmistaminen hankalaa (tietoja jää poimimatta) 	
Muut ongelmat	<ul style="list-style-type: none"> - Arkistoitavien tietojen erottelu lähdejärjestelmässä (työvoimaresurssien tarve kasvaa) 	

Ongelmien aiheuttamien riskien todennäköisyyksiä ei voida arvioida neljän projektin perusteella. Haasteet vaihtelevat suuresti projektien välillä, eikä aineistotoimituksen tyypikään kerro suoraan projektissa kohdattavista haasteista.

6. ONGELMAN RATKAISU

Luku kertoo havaittujen ongelmien ja haasteiden ratkaisemisesta. Tämän työn tavoitteena on selvittää arkistointiin liittyvät haasteet ja jäsentää käytetty prosessi paremmin. Luvun alussa esitetään kysymyksiä, joiden avulla arkistointiprojektin kulkua voidaan arvioida paremmin. Sen jälkeen esitetään ratkaisuja joihinkin ongelmiin, mutta ratkaisuja ei ole toteutettu eikä niiden toimivuutta voida arvioida. Lopuksi esitellään sisäisen tietomallin käyttöönotto, joka voi auttaa arkistointiprojekteissa tulevaisuudessa.

6.1 Alkuanalyysin kehittäminen

Seuraavassa esitetään vanhojen potilastietojen arkistointiprosessin tueksi kysymyksiä, joita olisi hyvä miettiä alkuanalyysin aikana. Näiden kysymysten avulla saadaan jonkinlainen käsitys arkistointiprojektin haastavuudesta ja mahdollisista riskeistä.

Onko järjestelmää arkistoitu ennen? Tunnetun järjestelmän arkistointi onnistuu helpommin, mutta silloinkin voi olla mahdollista, että järjestelmästä löytyy uusia ominaisuuksia, joita ei olla vielä huomioitu. Potilastietojärjestelmät ovat laajoja ohjelmistokokonaisuuksia, jotka toimitetaan räätälöityinä kokonaisuuksina asiakkaille, ja onkin tavallista, että ohjelmistoa on mukautettu käyttäjän tarpeiden mukaisesti. Tästä johtuen myös aikaisemmin arkistoitujen järjestelmien uudet versiot voivat sisältää uutta sisältöä tai ominaisuuksia, joita kaikki käyttäjät eivät ole käyttäneet, mistä johtuen kyseisiä ominaisuuksia ei ole vielä toteutettu arkistointiohjelmistoon.

Täysin tuntematon järjestelmä vaatii pahimmillaan huomattavan paljon aikaa, sillä alussa aikaa käytetään järjestelmään tutustumiseen ja toimintaperiaatteiden ymmärtämiseen. Kuten edellä mainittiin, potilastietojärjestelmät ovat räätälöitäviä kokonaisuuksia, joten alussa on hyvä koittaa selvittää, mitä ominaisuuksia on todennäköisesti kaikilla vastaavilla järjestelmillä. Samalla on tärkeää miettiä, mitä eroja tulevissa järjestelmissä saattaa olla, jottei ohjelmistoratkaisua tehdä liian jäykäksi. Mitä suurempi ohjelmistoratkaisu on kyseessä, sitä todennäköisempää on, että se on suhteellisen hyvin organisoitu ja noudattaa monilta osin samoja toteutuskäytäntöjä.

Kuinka suuri järjestelmä on kyseessä? Järjestelmän kokoa voidaan mitata esimerkiksi käyttäjämäärillä tai potilasmäärillä. Suuressa järjestelmässä on todennäköisesti esimerkiksi enemmän käyttäjiä, josta seuraa suuri määrä vaihtelevia kirjaustapoja. Myös

käyttäjien tavat käyttää järjestelmää vaihtelevat. Suuret potilasmäärät kasvattavat arkistoitavan tiedon määrää ja voivat myös kertoa siitä, että järjestelmä on ollut käytössä kauan, minkä vuoksi järjestelmästä voi löytyä päivitysten aiheuttamia virheitä.

Kuinka kauan järjestelmää on käytetty? Ohjelmistot ovat ylläpidettäviä kokonaisuuksia, joten pitkät käyttöjaksot tarkoittavat, että järjestelmää on mahdollisesti päivitetty. Päivityksissä järjestelmän tietokantaan saattaa jäädä sisältöä, jota ohjelmisto ei käytä. Tämänkaltaisen sisältö on ongelmallista, sillä kyseessä on potilastieto, jota ei voi hävittää. Toisaalta voi myös olla hyvin haastavaa tulkita haamuisältöä, jos ohjelmisto ei enää näytä sitä. Tällöin joudutaan käyttämään suuresti aikaa siihen, että mitä kukin tietokenttä tarkoittaa ja missä esitysmuodossa sisältö pitäisi esittää.

Pitkät käyttöjaksot tarkoittavat myös, että järjestelmään on saattanut tulla virheen korjauksia, jotka ovat muuttaneet jotakin toimintoa. Tämänkaltaisia asioita voi olla haastava havaita jälkeinpäin, mutta tiedostamalla asia jo aikaisessa vaiheessa, voidaan sen kanssa tulla toimeen.

Mitä tietoa järjestelmästä löytyy? Sisällön kartoittamiseksi luovutettua aineistoa tutkitaan analyysiympäristössä. Mikäli saatavilla on pääsy vain järjestelmän tietokantaan, analyysissä voidaan käydä läpi vain tietokerrosta, ja silloin suurten järjestelmien arkistoinnin kattavuuden varmistaminen voi olla kohtuuttoman haastavaa. Tämä johtuu siitä, että voi olla hyvin haastava varmistaa esimerkiksi numeroiksi koodattujen kenttien merkitys. Näissä tapauksissa asiakkaan asiantuntemuksella voi olla ratkaiseva merkitys.

Analysoinnin apuna voidaan käyttää myös alkuperäistä järjestelmää ja kaikkea dokumentaatiota mitä järjestelmästä löytyy. Dokumentaatiota ovat esimerkiksi versiopäivitysten muutosjulkaisut ja käyttöohjeet. Erityisesti käyttöohjeet tarjoavat yleensä hyödyllisiä vinkkejä, joita käyttäjätäkään eivät tunne. Käyttöohjeista saattaa myös löytyä termejä, joita tietokannassa on käytetty ja kuvia aikaisempien versioiden ohjelmaikkunoista.

Mitä tietoja siirretään? Rajanveto potilastiedon ja ei-siirrettävän tiedon välillä voi olla hyvin haastava, kuten aikaisemmissa luvuissa on kerrottu. Kun tiedetään, mitä tietoa järjestelmästä löytyy, voidaan miettiä, mikä tieto on potilastietoa, mikä muuten pakollista säilyttää ja mikä turhaa. Säilytettävää tietoa voivat olla esimerkiksi käyttölokit ja kirjanpitoon liittyvät asiat. Näistä on hyvä ainakin tiedottaa asiakasta.

Miten tieto irrotetaan? Potilastietojärjestelmät voivat sisältää hyvin monimuotoista sisältöä: kuvaajat (kasvukäyrät, verensokeriseurannat), hammaskartat, tekstikirjaukset, päivittäismerkinnät/seurannat, kuvat, liitteet. Tieto voi lisäksi olla säilöttynä useaan paikkaan. Järjestelmään on esimerkiksi saatettu liittää erillinen laboratorio-ohjelmisto tai kuvantamisiin käytettävä ohjelmisto. Silloin tietoa täytyy etsiä useammasta eri paikasta.

Tapa, jolla järjestelmä muodostaa tiedosta näytettävän sisällön voi olla hyvin monimutkainen esimerkiksi hammaskarttojen tai kasvukäyrien tapauksissa. Silloin vaihtoehtona on ohjelmistorobotiikan [41] käyttö, jossa erillinen ohjelma käyttää järjestelmää käyttäjän lailla ja irrottaa sieltä aineiston esimerkiksi kuvakaappauksilla tai kuvantunnistuksella. Mikäli se ei ole mahdollista, saatetaan järjestelmän toiminta joutua käänteismallintamaan, eli muodostamaan käsitys siitä, kuinka järjestelmä toimii. Tämän jälkeen sisältö voidaan muodostaa samalla tavalla kuin järjestelmä sen muodostaa, ja viedä sopivaan arkistointimuotoon. Voi olla kuitenkin hyvin haastavaa muodostaa tiettyjä sisältöjä identtisesti järjestelmän kanssa, esimerkiksi jos näytettävä muoto muodostetaan pelkästään ohjelman sisäisillä metodeilla.

Miten vanha tieto sovitetaan uuteen muotoon? Potilasasiakirjan tallentaminen asiakirjamuodossa onnistuu hyvin ja esimerkiksi päivittäismerkinnät voidaan kasata vuoden ajalta yhteen merkintään. Terveystieteidenhuollossa on usein käytössä erilaisia lomakkeita ja niitä käytetään myös sähköisissä järjestelmissä. Lomakkeet voidaan tallentaa esimerkiksi PDF-tiedostoina tai html:n avulla näyttömuodossa. PDF mahdollistaa esimerkiksi kasvukäyrien tallentamisen siten, että niitä voidaan lukea alkuperäisen kaltaisesti. Voi kuitenkin löytyä sisältöä, jonka tallentaminen asiakirjamuotoon vaatii paljon työtä tai se ei onnistu. Tämän kysymyksen tarkoituksena on varmistaa, että arkistoija on miettinyt, että kaikki tiedot sopivat asiakirjamuotoon.

6.2 Testauksen kehittäminen

Aliluvussa 5.7 kerrottiin, että testauksen suurimmat haasteet ovat suuri manuaalisen työn määrä ja automaattisen testauksen haastavuus. Projektin kehitys- ja testausvaiheessa testausta ja validointia tehdään samaan aikaan, sillä testauksen aikana aineistoa käydään läpi ja samalla saatetaan löytää tilanteita, joissa aikaisemmat oletukset eivät pidäkään paikkaansa, eli ohjelmisto toimii oikein, mutta lopputulos ei kuitenkaan ole haluttu.

Testaus joudutaan tekemään manuaalisesti, sillä virheitä etsitään vertaamalla transformoitua aineistoa alkuperäiseen aineistoon, jolloin virheiden tunnistaminen automaattisesti on hyvin haastavaa kuten aliluvussa 5.7 kerrotaan. Työn määrä on suuri, sillä testauksen kattavuus riippuu suoraan aineiston läpikäynnin kattavuudesta. Manuaalinen testaus on hidasta, aikaa vaativaa ja alttiimpaa virheille. Manuaalisen testauksen vaihtoehto on automaattinen testaus, jossa testausohjelmisto käy aineistoa läpi ja tarkistaa sääntöjen avulla vastaako se vaadittua.

Nykyisessä prosessissa testausta ja validointia voitaisiin siirtää ohjelmistokehitykseen käyttämällä erilaista ohjelmointityyliä. Sen sijaan, että ohjelmisto kehitetään toimimaan virheellisillä tai puutteellisilla syötteillä, virhetilanteisiin reagoidaan välittömästi, jolloin ne huomataan aikaisemmin. Voidaan esimerkiksi olettaa, että kaikilla käsiteltävillä potilailla on ainakin yksi vastaanottokäynti ja jos näin ei ole, siitä aiheutuu virhe. Virheen ansiosta voidaan huomata, että potilailla ei aina ole vastaanottokäyntiä, vaan jokin muu kirjaus. Testausta voidaan nopeuttaa myös sillä, että ohitetaan asiakirjan muodostusvaihe. Muodostuksessa tapahtuu yleensä vähemmän virheitä, joten testaaminen pelkän muunnosvaiheen avulla nopeuttaisi virheiden löytämistä ja mahdollistaisi aineiston testaamisen säännöllisesti prosessin aikana.

Edellä kuvattu virhe olisi pitänyt löytää jo aineiston analyysivaiheessa. Migraatioprosessin kuvauksessa luvussa 2.1.1 kerrotaan profilointityökalusta, jolla aineiston laatua ja ominaisuuksia tutkitaan migraatioprosessin alussa. Työkaluun voidaan määrittää oletukset, jotka aineiston pitäisi täyttää ja näin oltaisiin voitu löytää edellä kuvattu virhe. Nykyisessä prosessissa analyysivaiheessa ei kuitenkaan käytetä kuvattua työkalua, vaan kaikki virhetilanteet pyritään löytämään testauksessa. Validointityökalun käyttö alkuvaiheessa voisi auttaa löytämään virheitä tehokkaammin ja samassa vaiheessa muodostettava tietomallista olisi hyötyä myös seuraavissa prosessin vaiheissa. Eräänlainen tietomalli esitetään luvussa 6.4.

6.3 Datamigraation kattavuus

Datamigraation kattavuuden varmistaminen on erittäin tärkeä ongelma, sillä potilastietoa ei saisi hävitä arkistoinnissa. B-projektissa käytettiin menetelmää, jossa tietojen käsittely varmistettiin aineiston kokonaismäärän avulla. Yhtenä vaihtoehtona voisi olla myös väli­muodon käyttö poiminnan jälkeen, jolloin poiminnan validointi voisi olla helpompaa.

B-projektissa otettiin käyttöön rivimääriin perustuva menetelmä käsittelyn kattavuuden varmistamiseksi. Menetelmässä käsiteltyjen tietokantarivien määrät kirjattiin, jolloin niitä voitiin verrata tietokannan rivimääriin. Tietokannan rivimäärät saatiin muodostamalla tietokantakyselyt, jotka laskivat niiden rivien määrät, jotka arkistoinnin piti käsitellä. Menetelmän haasteena oli tilanteet, joissa sama rivi käsiteltiin useasti, jolloin käsiteltyjen rivien määrä näytti suuremmalta kuin se todellisuudessa oli. Myös tietokantakyselyiden muodostaminen oli haastavaa, sillä poimintavaiheessa tietokannasta luettiin lähes kaikki tiedot ja tietoja suodatettiin ohjelmallisesti, joten kyselyinä ei voitu käyttää poiminnassa käytettyjä kyselyitä.

Menetelmän käyttö auttoi löytämään tilanteita, jossa poiminta suodatti liikaa tietoa. Tilanteet johtuivat esimerkiksi tietokannan taulujen välisten relaatioiden väärintulkinnasta. Yhdessä tilanteessa tietokannan taulujen välillä oli yksi-moneen viittaus yksi yhteen viittauksen sijaan. Menetelmän avulla huomattiin, että osa tiedoista jäi käsittelemättä ja virhe saatiin korjattua. Virhe oli niin vähäinen, että manuaalisella testauksella sitä ei todennäköisesti olisi löydetty.

Ratkaisuna voisi olla myös seuraavassa aliluvussa käsiteltävä väliaikainen poiminta-alue, johon käsiteltävä tieto haetaan. Sen avulla voitaisiin helpommin varmistaa, että ainakin kaikki tarpeellinen on poimittu, sillä tietoa ei pitäisi joutua käsittelemään useaan kertaan. Myös poiminnan validointi voisi olla helpompaa poiminnan ja transformaatioiden erottamisen vuoksi.

Ongelmaan ei ole kuitenkaan vielä keksitty ratkaisua, jolla arkistoinnin kattavuudesta voitaisiin olla täysin varmoja. Tällä hetkellä varmistaminen vaatii vielä tarkkaa manuaalista työskentelyä ja asiakkaan osallistumista.

6.4 Sisäisen tietomallin käyttö

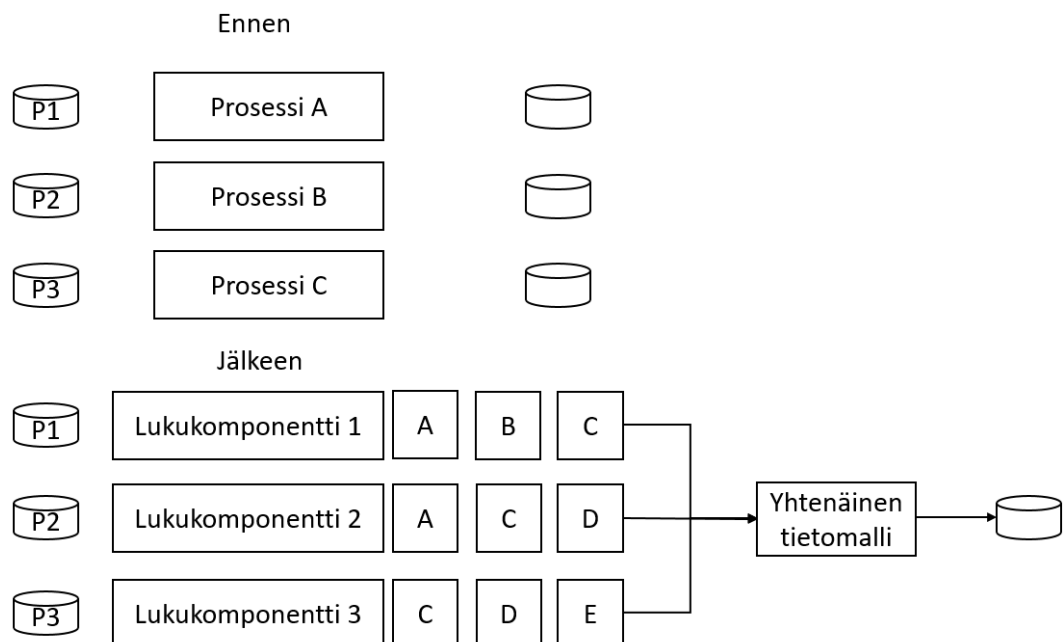
Luvussa 2.1 käsitellyssä ETL-prosessin transformaatiovaiheessa, transformaatiot tapahtuivat lähde- ja kohdetietomallien kenttien välillä. Transformaatioissa voitiin esimerkiksi vaihtaa kentät tietotyyppi tai hakea kentän arvolle arvo toisen kentän perusteella. Vanhojen potilastietojen arkistointiprosessissa, transformaatiota ei suoriteta tietokenttien välillä, vaan lähdejärjestelmän kenttien ja muodostettavan asiakirjan välillä.

Transformaatioiden muodostaminen ja uudelleenkäyttäminen tässä tapauksessa on haastavampaa, sillä transformaatio on monimutkaisempi, kuin kenttien välissä tapahtuvassa transformaatioissa. Transformaatioiden kehittäminen olisi helpompaa, jos ne tapahtuisivat kenttien välillä.

Arkistoinnin prosessi on usein sisällöltään hyvin samankaltainen projektien välillä. Lähtötietoina on potilastiedot, jotka ovat sisällöllisesti samankaltaisia. Tiedoista koostetaan asiakirjoja, jotka arkistoidaan potilastiedon arkistoon. Lähtötietojen samankaltaisuudesta huolimatta, projekteissa niitä käsiteltiin täysin omina prosesseinaan. Transformaatioita voitaisiin käsitellä kenttien välillä, jos lähdejärjestelmän tietokentän ja muodostettavan asiakirjan välille kehitetään tietomalli, jonka perusteella asiakirja muodostetaan. Prosessissa potilastiedot poimitaan lähdejärjestelmästä, suoritetaan transformaatio tietomalliin ja toinen transformaatio tietomallista asiakirjaksi.

Tietomalli olisi rakenteeltaan sellainen, että siihen voitaisiin lukea kaikki potilastiedot mitä lähdejärjestelmistä löytyy. Tietomalliin pitäisi pystyä tallentamaan esimerkiksi tekstimuotoisia kirjauksia, mittaustuloksia, PDF-tiedostoja ja muotoiltua tekstiä. Tietomallia voidaan laajentaa, kun tietojen arkistoinnissa tulee vastaan tietoa, joka ei sovi tietomalliin. Näin tietomalli paranee ajan kuluessa ja seuraavat potilastietojen arkistoinnit voidaan suorittaa helpommin.

Sisäisen tietomallin lisääminen prosessiin muuttaa sen poimintavaihetta ja transformaatiovaihetta. Poimintavaiheessa tiedot poimitaan lähtöjärjestelmästä sisäiseen tietomalliin, käyttäen sopivia transformaatiokomponentteja. Komponentit voisivat esimerkiksi koostaa tietokannan taulun riveistä yhtenäisen tekstin tai hakea tietokannan kentän perusteella arvon toisesta tietokannan taulusta. Transformaatiovaiheessa tietomallista muodostetaan potilasasiakirjat, jotka ladataan potilastiedon arkistoon.



Kuva 11. Uudistettu ohjelmistoratkaisu.

Kuva 11 esittää vanhaa ja uudistettua ohjelmiston näkökulmasta. Kuvan yläosassa vasemmalla on kolme potilastietojärjestelmää, joista tiedot luetaan ja käsitellään omissa prosesseissaan ja lopputuloksena syntyy kolme aineistoa. Alaosan uudistetussa prosessissa jokaista lähdejärjestelmää kohden on luotu oma lukukomponenttinsa, joka lukee järjestelmän tiedot tietomalliin, joka on sama kaikissa projektissa. Lukukomponentit käyttävät transformaatioita A–E, jotka suodattavat, validoivat tai rikastavat tietoa. Transformaatiot ovat yhteisiä kaikille järjestelmille ja niitä voidaan kehittää ja käyttää toisistaan

riippumatta. Kun tiedot on luettu tietomalliin, niistä muodostetaan omalla transformaatiolla asiakirjoja, jotka ladataan potilastiedon arkistoon.

Sisäisen tietomallin käytöstä on hyötyä myös testauksessa ja kehityksessä. Luvussa 2.1.1 kerrottiin välivarastosta, johon poimittu aineisto voidaan tallentaa. Kun tietomalli tallennetaan aineiston poiminnan jälkeen, virhetilanteen sattuesssa aineistoa ei tarvitse poimia uudestaan, vaan prosessia voidaan jatkaa edellisellä tietomallilla. Toisaalta, jos poiminta-algoritmissa havaitaan virhe, poimittu aineisto on osittain tai kokonaan virheelinen, joten sitä ei voida käyttää.

Myös transformaatioiden validointi voisi olla helpompaa tietomallia käyttämällä. Uudessa prosessissa transformaatiot kohdistuisivat lähde- ja kohdetietomallien kenttien välille, lähdejärjestelmän ja kohdeasiakirjan sijaan. Tietomallien väliset transformaatiot olisivat yksinkertaisempia muodostaa, mutta myös niiden tarkastaminen ja validointi olisi helpompaa. Lisäksi transformaatioiden kehittämiseksi voitaisiin luoda oma kuvauskieli, jolloin niiden kehitykseen ja validointiin voisi osallistua myös asiakas.

7. YHTEENVETO

Työn tarkoituksena oli dokumentoida ja analysoida vanhojen potilastietojen arkistointiprojekteissa käytetty prosessi sekä projekteihin liittyvät haasteet ja riskit. Näiden perusteella pyrittiin kehittämään prosessin ennustettavuutta ja riskienhallintaa tulevia projekteja varten. Työssä analysoitiin neljä eri vanhojen potilastietojen arkistointiprojektia, joissa siirrettiin potilastiedot vanhasta potilastietojärjestelmästä kansalliseen potilastiedon arkistoon.

Kansallinen potilastiedon arkisto on Kelan tuottama lain velvoittama tietojärjestelmäpalvelu, johon yksityiset ja julkiset terveydenhuollon palveluntarjoajat voivat liittyä ja siirtää vanhat ja liittymisen jälkeen syntyneet potilastiedot pitkäaikaissäilytykseen. Arkisto vastaa sen jälkeen tietojen lainmukaisesta säilyttämisestä säilytysajoista asetetun asetuksen vaatimusten mukaisesti. Näin ollen palvelun käyttäjän ei enää tarvitse huolehtia potilastietojen säilytyksestä. Lisäksi tietojen käyttäminen voi helpottua, sillä tiedot ovat laajemmin käytettävissä.

Aiemmin ongelmana on ollut, että jos terveydenhuollon palveluntarjoaja on halunnut vaihtaa potilastietojärjestelmää, on vaihtoehtoina ollut joko potilastietojen siirtäminen vanhasta järjestelmästä uuteen järjestelmään tai vanhan järjestelmän säilyttäminen uuden järjestelmän rinnalla. Ensimmäiseen liittyy riski epäonnistuneesta migraatiosta, eli tiedon siirtämisestä järjestelmästä toiseen ja toisen suurin ongelma on kahden järjestelmän yhtäaikaisen ylläpidon aiheuttamat kustannukset. Ratkaisuna näihin ongelmiin, vanhan potilastietojärjestelmän potilastiedot voidaan siirtää potilastiedon arkistoon ja vanhasta potilastietojärjestelmästä voidaan luopua.

Työssä vanhojen potilastietojen arkistointia tarkasteltiin datamigraatioprosessina, joka koostuu aineiston toimituksesta, alkuanalysista, iteratiivisesta transformaatiovaiheesta, testauksesta ja validoinnista sekä tuotantoonviennistä. Prosessin lopputuotteena vanhan potilastietojärjestelmän potilastiedot on muunneltu kansallisen potilastiedon arkiston vaatimaan asiakirjamuotoon. Kela lataa ne lopuksi potilastiedon arkistoon.

Prosessille asettavat reunaehdot laista tulevat vaatimukset, kelan tekniset määräykset asiakirjojen muodosta ja THL:n määräykset. Tavallisestikin laki tai viranomaiset saattavat asettaa datamigraatioprosesseille vaatimuksia, mutta potilastietojen arkistoinnissa laista ja viranomaisilta tulevat vaatimukset vaikuttavat hyvin suuresti projektin läpivientiin.

Käsittelyt projektit jaettiin aineiston toimitustavan mukaan kahteen tyyppiin: valmiiksi poimitut ja poimimattomat aineistot. Poimimattomat aineistot tarkoittavat esimerkiksi aineistotoimitusta, joka koostuu potilastietojärjestelmän tietokannasta tai potilastietojärjestelmästä. Aineiston käsittelemiseksi sen sisältö täytyy ymmärtää. Tämän takia aineistoa analysoidaan esimerkiksi selvittämällä sen tietokannan rakennetta tai käyttämällä potilastietojärjestelmää ja tutkimalla sen toimintaa.

Poimittu aineisto tarkoittaa, että asiakas toimittaa aineiston ja kuvauksen aineiston sisällöstä ja tulkinnasta. Tällöin järjestelmän analysointi on jo tehty ja löydöt dokumentoitu. Tämä toimitusmuoto säästää arkistojen monelta työvaiheelta, mutta saattaa aiheuttaa asiakkaalle lisää kustannuksia ja aiheuttaa riippuvuuden kolmanteen osapuoleen, joka poimii aineiston. Riippuvuus kolmannelta aiheuttaa riskin projektin keston pitenemiselle ja mahdollisuuden, että potilastietoja jää arkistoimatta. Kun aineiston poiminnan suorittaa alkuperäisen järjestelmän tunteva kolmas osapuoli, riski aikataulun venymiselle pienenee ja tulkintavirheiden todennäköisyys lähes häviää. On kuitenkin mahdollista, että järjestelmä on niin vanha, ettei edes järjestelmän tunteva kolmas osapuoli, pysty poimaan aineistoa.

Kun aineisto toimitetaan valmiiksi poimittuna, asiakas vastaa siitä, että aineisto on poimittu oikein. Vaikka aineisto toimitettaisiin poimimattomana, asiakas vastaa myös silloin sen arkistoinnista, mutta arkistojalla on silloin enemmän mahdollisuuksia varmistaa, että kaikki potilastiedot arkistoidaan, eikä asia jää ainoastaan asiakkaan testattavaksi.

Poimimattomissa aineistoissa on mahdollista, että poimintaa ei voida itse tehdä, vaan asiakkaan on tilattava se kuitenkin kolmannelta osapuolelta. Näin voi olla esimerkiksi tilanteessa, jossa järjestelmän tietokanta on niin monimutkainen, ettei sitä voida ymmärtää. Monimutkaisesta tai järjestelmästä huolimatta, poiminta voidaan joskus tehdä, mutta silloin on riski, että potilastietoja jää arkistoimatta tai niitä arkistoidaan väärin. Myös projektin aikataulu saattaa pidentyä ja henkilöresurssitarve kasvaa.

Tutkittavasta järjestelmästä saattaa löytyä tietoa, joka ei muodoltaan sovi arkistoitavaksi. Tällaisia tietoja ovat esimerkiksi videot ja ääninauhoitteet. Lisäksi esimerkiksi interaktiivisten näyttömuotojen muuntaminen arkistoitavaan muotoon voi kasvattaa henkilöresurssitarvetta.

Työssä ei käsitelty potilastiedon arkistoon ladatun virheellisen aineiston korjaamista, sillä käsitellyissä projekteissa ladatussa aineistossa ei ole havaittu virheitä. Potilastiedon arkistossa on kuitenkin huomioitu mahdollisuus virheiden korjaamiselle, mutta näissä projekteissa sitä ei ole käytetty. Käsitellyissä projekteissa ei ole varauduttu virheen mahdollisuuteen eikä virheellisten asiakirjojen käsittelyä ole huomioitu prosessissa. Projektien

aineistot on kuitenkin jo arkistoitu kantaa, joten alkuperäistä aineistoa ei enää välttämättä ole ja virheiden korjaaminen voi olla mahdotonta. Jatkossa asiakas voisi tallentaa arkistoidun aineiston määrääjäksi, jolloin virheitä voitaisiin korjata, jos niitä huomataan arkistoinnin jälkeen. Toisaalta vanhoja potilastietoja ei välttämättä käytetä hoitotilanteissa aktiivisesti, jolloin myös virheiden löytymisen todennäköisyys on pieni.

Käsitellyissä projekteissa migraation onnistuminen varmistetaan vain pienen potilasosajoukon tarkastamisella. Arkistoitavia potilaita voi olla yli 100 000, mutta arkistoinnin tarkistuksessa niistä tarkistetaan vain kymmeniä, eli alle 0,1 prosenttia. Tarkastuksen tekee asiakas. Lisäksi aineiston automaattinen testaus on haastavaa, sillä arkistointimuotoon muunnettu aineisto eroaa hyvin paljon lähtöaineistosta, joten tarkistussääntöjen muodostaminen on haasteellista.

Arkistointiprosessin kehittäminen on haastavaa, sillä vaikka prosessin päämäärä on aina sama, projektit eroavat aina toisistaan. Projekteissa kohdattavat ongelmat ovat hyvin yksilöllisiä, eikä niiden ratkaisuja voida yleistää muihin projekteihin. Kuitenkin lähtöaineistot ovat yleensä samankaltaisia, joten niitä voidaan analysoida samoilla menetelmillä.

Hyvin tehdyllä alkuanalyysillä projektin kesto ja henkilötyöresurssien tarve voidaan arvioida tarkemmin ja projektiin liittyviin riskeihin voidaan varautua aikaisemmin. Sen tähden merkittävänä osana ongelman ratkaisua on esitetty kysymyksiä alkuanalyysin jäsentämiseksi. Arkistointiin liittyviä riskejä voidaan kartoittaa muun muassa luvusta 6 poimituilla kysymyksillä:

- Onko järjestelmää siirretty ennen?
- Kuinka suuri järjestelmä on kyseessä?
- Kuinka kauan järjestelmää on käytetty?
- Mitä tietoa järjestelmästä löytyy?
- Mitä tietoja siirretään?
- Miten tieto irrotetaan?
- Miten vanha tieto sovitetaan uuteen muotoon?

Alun perin ohjelmistoratkaisu kehitettiin siten, että kaikkia arkistoitavia potilastietojärjestelmiä varten kehitetään oma ohjelmistokomponentti, joka lukee potilastietojärjestelmien potilastiedot ja muodostaa niistä asiakirjoja. Projekteissa on kuitenkin havaittu, että potilastietojärjestelmien potilastiedot sisältävät samankaltaisuuksia, joten voisi riittää, että potilastiedot luetaan ohjelmiston sisäiseen tietomalliin, jonka jälkeen ohjelmiston runko muodostaa niistä asiakirjoja. Erona nykyiseen ratkaisuun on, että asiakirjoja ei tarvitse

aina muodostaa uudelleen jokaisessa ohjelmistokomponentissa, vaan voidaan uudelleenkäyttää jo kerran luotuja ohjelmistokomponentteja, jotka käsittelevät tietynlaisia potilastietoja. Hyötynä tästä on se, että ohjelmistokehityksen määrä pienenee, jolloin aikaa voidaan käyttää enemmän testaukseen.

Prosessia voisi myös parantaa erillisellä aineistopoiminnalla, jossa lähdejärjestelmästä poimitut potilastiedot tallennetaan välivarastoon. Nykytilanteessa poiminta suoritetaan prosessin aikana ennen muunnosta, mutta tietoja ei tallenneta mihinkään. Välivarastoinnin etuna on, että poimittua aineistoa on helpompi verrata automaattisesti lähdejärjestelmän tietokantaan. Lisäksi se voi nopeuttaa prosessia, sillä prosessin uudelleenajojen välillä aineistoa ei tarvitse aina poimia uudelleen lähdejärjestelmästä. Toisaalta välivarastoitua aineistoa ei välttämättä voida käyttää, sillä kehityksessä on yleistä, että poiminta-algoritmia muokataan.

Työ dokumentoi projekteissa käytetyn datamigraatioprosessin haasteineen. Työssä annettiin myös prosessin vaiheille kehitysehdotuksia, prosessin kehittäminen on haasteellista, sillä projektien haasteet ovat aina yksilöllisiä. Erityisesti testauksen automatisointia pitäisi vielä kehittää. Työssä esiteltyjä kehitysehdotuksia ei ole vielä kokeiltu arkistointiprojekteissa, joten niiden toimivuutta ei voida arvioida.

LÄHDELUETTELO

- [1] Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä, "9.2.2007/159," [Online]. Available: <https://www.finlex.fi/fi/laki/ajantasa/2007/20070159>.
- [2] Laki potilaan asemasta ja oikeuksista, "17.8.1992/785," [Online]. Available: <https://www.finlex.fi/fi/laki/ajantasa/1992/19920785>.
- [3] Sosiaali- ja terveysministeriön asetus potilasasiakirjoista, "298/2009," [Online]. Available: <https://www.finlex.fi/fi/laki/alkup/2009/20090298>.
- [4] B. Research, "Data migration," 2011.
- [5] A. Haikonen, *Potilastietojen standardointi ja siirto kansalliseen sähköiseen arkistoon*, 2017.
- [6] B. Thalheim ja Q. Wang, "Data migration: A theoretical perspective," *Data & Knowledge Engineering*, osa/vuosik. 87, pp. 260-278, 2013.
- [7] J. Bisbal, "An overview of legacy information system migration," tekijä: *Proceedings of Joint 4th International Computer Science Conference and 4th Asia Pacific Software Engineering Conference*, Hong kong, 1997.
- [8] G. Canfora, A. R. Fasolino, G. Frattolillo ja P. Tramontana, "A wrapping approach for migrating legacy system interactive functionalities to service Oriented Architecture," *Journal of System and Software*, osa/vuosik. 81, nro 4, pp. 463-480, 2008.
- [9] R. Kimball ja J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data*, John Wiley & Sons, Inc., 2004.
- [10] Bloor Research, "Lower your risk with application data migration - next steps with Informatica," 2013.
- [11] D. Dolezel ja J. Moczygemba, "Implementing EHRs: An Exploratory Study to Examine Current Practices in Migrating Physician Practice," *Perspectives in Health Information Management*, 2015.
- [12] T. Bentley, M. Rizer, A. S. McAlearney, H. Mekhjian, M. Siedler, K. Sharp, P. Teater ja T. Huerta, "The journey from precontemplation to action: Transitioning between electronic medical record systems," *Health Care Management review*, osa/vuosik. 41, nro 1, pp. 22-31, 2016.
- [13] N. J. Girard, "DRESSed for Failure," *AORN Journal*, osa/vuosik. 101, nro 4, 2015.
- [14] J. Michel, A. Hsiao ja A. Fenick, "Using a Scripted Data Entry Process to Transfer Legacy Immunization Data While Transitioning Between Electronic Medical Record Systems," *Applied Clinical Information*, osa/vuosik. 5, nro 1, pp. 284-298, 2014.
- [15] A. Gettinger ja A. Csatari, "Transitioning from a Legacy EHR to a Commercial, Vendor-supplied, EHR," *Applied Clinical Informatics*, osa/vuosik. 3, nro 4, pp. 367-376, 2012.
- [16] S. Bornstein, "An integrated EHR at Northern California Kaiser Permanente: pitfalls, challenges, and benefits experienced in transitioning," *Applied Clinical Information*, osa/vuosik. 3, nro 3, pp. 318-325, 2012.
- [17] M. Makar, *Dealing with existing data in legacy systems when transitioning between Electronic Health Records in three Swedish counties*, Stockholm, 2014.
- [18] Sosiaali- ja terveysministeriö, "Potilasasiakirjojen laatiminen ja käsittely," 2012.
- [19] Arkistolaitos, *Potilastietojen sähköinen säilyttäminen Kelan Potilastiedon arkistossa*, 2015.
- [20] *General Data Protection Regulation*.
- [21] Kanta, "Omakanta," [Online]. Available: <https://www.kanta.fi/omakanta>.
- [22] Kela, "Vanhojen potilastietojen arkistointi," 8 8 2018. [Online]. Available: <https://www.kanta.fi/jarjestelmakehittajat/vanhojen-potilastietojen-arkistointi>. [Haettu 3 10 2018].
- [23] THL, "Sanasto," [Online]. Available: <https://thl.fi/fi/web/tiedonhallinta-sosiaali-ja-terveysalalla/koodistopalvelu/mika-koodistopalvelu-on/sanasto>.

- [24] HL7, "HL7," [Online]. Available: <https://www.hl7.org/>.
- [25] Kanta, "HL7-määrittelyt," [Online]. Available: <https://www.kanta.fi/jarjestelmakehittajat/hl7>. [Haettu 3 10 2018].
- [26] HL7, "CDA® Release 2," [Online]. Available: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=7.
- [27] Kanta, "Potilastiedon arkisto," 8 8 2018. [Online]. [Haettu 3 10 2018].
- [28] Kela, "Potilastiedon arkiston Kertomus ja lomakkeet 5.11," Kela, 2015.
- [29] Kela, "Kansallinen koodistopalvein," [Online]. Available: <https://koodistopalvelu.kanta.fi>.
- [30] Kela, "Vanhan aineiston tekniset vaatimukset versio 1.22," 2018.
- [31] Wikipedia, "Windows 2000," [Online]. Available: https://en.wikipedia.org/wiki/Windows_2000.
- [32] Kanta, "Kuva-aineistojen arkisto," [Online]. Available: <https://www.kanta.fi/jarjestelmakehittajat/kuva-aineistojen-arkisto>.
- [33] Kanta, *Kanta – Kertomus ja lomakkeet CDA R2-rakenne*, Kanta.
- [34] Kanta, "Vanhojen potilastietojen validoinnin testaustyökalu," [Online]. Available: <https://www.kanta.fi/jarjestelmakehittajat/vanhojen-potilastietojen-arkistointi>.
- [35] P. NM, "A rational approach to legacy data validation when transitioning between electronic health record systems," *Journal of the American Medical Information Association*, osa/vuosik. 23, nro 5, pp. 991-994, 2016.
- [36] "wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/File_Transfer_Protocol#Simple_File_Transfer_Protocol.
- [37] Wikipedia, "Microsoft SQL Server," [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_SQL_Server.
- [38] E. Chikofsky ja J.H.Cross, "Reverse engineering and design recovery: a taxonomy," *IEEE Software*, nro 7, pp. 13-17, 1990.
- [39] W. Premarlani ja M. Blaha, "An approach for reverse engineering relational databases," *Communications of the ACM*, osa/vuosik. 37, nro 5, 1994.
- [40] PDF association, "ISO 19005 (PDF/A)," [Online]. Available: <https://www.pdfa.org/resource/iso-19005-pdfa/>.
- [41] Wikipedia, "Robotic process automation," [Online]. Available: https://en.wikipedia.org/wiki/Robotic_process_automation.